

**A Thesis Submitted for the Degree of PhD at the University of Warwick**

**Permanent WRAP URL:**

<http://wrap.warwick.ac.uk/126000>

**Copyright and reuse:**

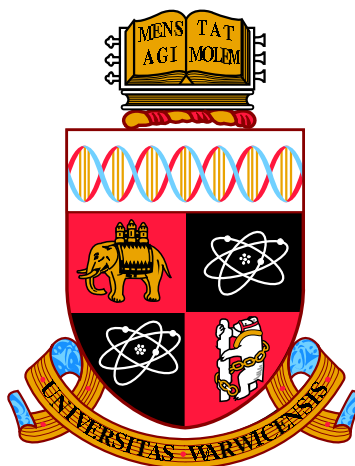
This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it.

Our policy information is available from the repository home page.

For more information, please contact the WRAP Team at: [wrap@warwick.ac.uk](mailto:wrap@warwick.ac.uk)



**Tool development for metabolic analyses in the  
context of thermodynamic constraints**

by

**Andrea Sofia Martinez Vernon**

**Thesis**

Submitted to the University of Warwick

for the degree of

**Doctor of Philosophy**

**School of Life Sciences**

September 2018

THE UNIVERSITY OF  
**WARWICK**



# Contents

<b>List of Tables</b>	<b>viii</b>
<b>List of Figures</b>	<b>x</b>
<b>List of Code</b>	<b>xiv</b>
<b>Acknowledgments</b>	<b>xv</b>
<b>Declarations</b>	<b>xvii</b>
Publications . . . . .	xviii
Funding . . . . .	xviii
<b>Abstract</b>	<b>xix</b>
<b>Abbreviations</b>	<b>xx</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Thermodynamics . . . . .	1
1.2 Metabolism and thermodynamics . . . . .	4
1.3 Metabolic-based interactions: syntrophy . . . . .	8
1.3.1 Syntrophic interactions and interspecies hydrogen transfer (IHT) . . . . .	10
1.3.2 Syntrophy case study: DvMm coculture . . . . .	10
1.3.3 Syntrophy and direct interspecies electron transfer (DIET) . . . . .	11
1.4 Electric interactions between conductive materials and microbes . . . . .	12
1.4.1 Electroactive microorganisms and extracellular electron transfer (EET) . . . . .	12
1.5 Bioelectrochemical systems (BES) . . . . .	13
1.6 Challenges remaining with electronic interfacing of microorganisms . . . . .	15
1.7 Aims and objectives . . . . .	15
1.7.1 “Syntrophy over wires” hypothesis . . . . .	16

1.7.2	Beyond a hypothesis: need for a computational tool for the discovery and design of electronic microbial interactions . . . . .	17
-------	--	----

## Chapter 2 Development of an electrochemical platform

	<b><i>Electrochemistry experiments with slow growing, anaerobic microorganisms</i></b>	<b>19</b>
2.1	Introduction . . . . .	19
2.2	Results . . . . .	22
2.2.1	Working and counter electrodes . . . . .	24
2.2.2	Reference electrode . . . . .	25
2.2.2.1	Reference electrode characterisation . . . . .	26
2.2.3	Electrochemical cell . . . . .	27
2.2.4	Anaerobic container . . . . .	29
2.2.5	System overview . . . . .	33
2.3	Discussion . . . . .	36
2.4	Materials and Methods . . . . .	39
2.4.1	Electrochemical measurements . . . . .	39
2.4.1.1	Potentiostat connections . . . . .	39
2.4.2	Adhesives . . . . .	40
2.4.3	3D printing . . . . .	40
2.4.4	Soldering . . . . .	40
2.4.5	Conductivity test . . . . .	40
2.4.6	Electrochemical cell . . . . .	41
2.4.6.1	Addition of pipe adaptors . . . . .	41
2.4.7	Rubber gasket production . . . . .	41
2.4.8	Cellulose membrane production . . . . .	42
2.4.9	Working and counter electrode manufacturing . . . . .	42
2.4.9.1	Gold coating of electrodes . . . . .	43
2.4.9.2	Counter electrode (CE) manufacturing . . . . .	43
2.4.9.3	Electrode support and separation . . . . .	44
2.4.10	Reference electrode manufacturing protocol . . . . .	44
2.4.10.1	4 M KCl electrolyte solution . . . . .	45
2.4.10.2	Glass capillary preparation . . . . .	45
2.4.10.3	Silver wire preparation . . . . .	45
2.4.10.4	Reference electrode assembly . . . . .	45
2.4.11	Reference electrode characterisation . . . . .	45
2.4.12	Production of the <i>modified stopper</i> . . . . .	46

2.4.13	Electrochemical cell assembly . . . . .	46
2.4.13.1	CE assembly . . . . .	47
2.4.13.2	WE assembly . . . . .	47
2.4.13.3	Gasket and membrane ‘sandwich’ assembly . . . . .	47
2.4.13.4	Electrochemical cell assembly . . . . .	47
2.4.14	Electrochemical cell sterilisation and degassing . . . . .	48
2.4.15	Anaerobic container preparation . . . . .	48
2.4.15.1	Container anaerobicity test . . . . .	48
2.4.15.2	External container connections . . . . .	48
2.4.15.3	Internal container connections . . . . .	50
2.4.15.4	Electrical noise elimination . . . . .	50
2.4.16	Connections required for ZRA measurements . . . . .	51
2.4.17	Experimental platform set-up . . . . .	51
<b>Chapter 3</b>	<b>Towards investigating the “syntrophy over wires” hypothesis</b>	<b>53</b>
3.1	Introduction . . . . .	53
3.2	Results . . . . .	58
3.2.1	Current . . . . .	58
3.2.1.1	Estimation of the number of electrons exchanged . . . . .	60
3.2.2	Biofilm characterisation . . . . .	61
3.2.3	Characterisation of catalytic reactions . . . . .	61
3.2.4	Characterisation of Mm’s metabolic activity . . . . .	66
3.2.5	Characterisation of Dv’s metabolic activity . . . . .	67
3.2.6	Bacterial growth . . . . .	69
3.3	Discussion . . . . .	71
3.4	Materials and Methods . . . . .	77
3.4.1	Experimental design . . . . .	77
3.4.2	Microorganisms and culturing . . . . .	78
3.4.2.1	Microorganism-specific additives . . . . .	79
3.4.2.2	Cryostock production . . . . .	79
3.4.2.3	Seed cultures . . . . .	79
3.4.2.4	Hungate tube cultures . . . . .	79
3.4.2.5	DvMm.O2 tubes . . . . .	80
3.4.3	Theoretical gas production calculation . . . . .	80
3.4.3.1	Calculating the maximum amount of methane that can be produced	80
3.4.3.2	Calculating the maximum methane volume that can be produced	80

3.4.4	Electrochemical cell assembly . . . . .	80
3.4.5	Electrochemical cell inoculation . . . . .	81
3.4.6	Electrochemical system . . . . .	81
3.4.6.1	Summary of electrochemical cell arrangement . . . . .	82
3.4.7	Equipment used for electrochemical measurements . . . . .	82
3.4.8	Current measurement . . . . .	82
3.4.8.1	Current integration over time . . . . .	83
3.4.9	Electrochemical Impedance Spectroscopy (EIS) measurement . . . . .	84
3.4.10	Cyclic voltammetry (CV) measurement . . . . .	85
3.4.10.1	Mid-peak potential estimation . . . . .	87
3.4.11	Gas chromatography . . . . .	88
3.4.12	Ion chromatography . . . . .	89
3.4.12.1	Calibration standard solution preparation . . . . .	89
3.4.12.2	Sample preparation . . . . .	89
3.4.12.3	Separation protocol . . . . .	89
3.4.12.4	Calibration . . . . .	90
3.4.13	Optical density (OD <sub>600</sub> ) . . . . .	90
3.4.14	Statistical analyses . . . . .	91

## Chapter 4 MetQy

	<i>An R package to query metabolic functions of genes and genomes</i>	<b>92</b>
4.1	Introduction . . . . .	92
4.2	MetQy and the use of KEGG data . . . . .	94
4.2.1	KEGG orthology data . . . . .	94
4.2.2	KEGG enzyme data . . . . .	95
4.2.3	KEGG genome data . . . . .	95
4.2.4	KEGG module data . . . . .	96
4.2.4.1	KEGG module definition . . . . .	97
4.3	MetQy package description . . . . .	97
4.3.1	MetQy parsing functions . . . . .	99
4.3.1.1	<i>parseKEGG_file</i> . . . . .	99
4.3.1.2	<i>parseKEGG_file.list</i> . . . . .	99
4.3.2	MetQy query functions . . . . .	100
4.3.2.1	<i>query_genomes_to_modules</i> . . . . .	101
4.3.2.2	<i>query_modules_to_genomes</i> . . . . .	103
4.3.2.3	<i>query_genes_to_modules</i> . . . . .	103

4.3.2.4	<i>query_genes_to_genomes</i>	104
4.3.2.5	<i>query_missingGenes_from_module</i>	104
4.3.3	MetQy analysis functions	105
4.3.3.1	<i>analysis_pca_mean_distance_calculation</i>	106
4.3.3.2	<i>analysis_pca_mean_distance_grouping</i>	107
4.3.3.3	<i>analysis_genomes_module_output</i>	108
4.3.4	MetQy visualisation functions	108
4.3.4.1	<i>plot_heatmap</i>	109
4.3.4.2	<i>plot_scatter</i>	109
4.3.4.3	<i>plot_scatter_byFactors</i>	111
4.3.4.4	<i>plot_sunburst</i>	112
4.3.4.5	<i>plot_variance_boxplot</i>	114
4.4	Using MetQy: a biological example	114
4.4.1	Identify potential methanogens and evaluate their genomes for selected metabolic processes loosely relating to anaerobic digestion.	115
4.4.2	Visualise the <i>mcf</i> for the user-specified modules across the selected genomes in KEGG using a heatmap	117
4.4.3	Investigate which are the genomes that have a low <i>mcf</i> for module ‘M00567’ based on the heatmap	118
4.4.4	Find out which genes are missing for module ‘M00567’ to be complete for genome ‘T04272’	118
4.4.5	Use a sunburst plot to visualise ‘T04272’s <i>mcf</i> for all the modules including module information.	119
4.4.6	Carry out an automated analysis	120
4.5	Discussion	126

## Chapter 5 Identification of anodic and cathodic organisms

	<i>Towards testing the “syntrophy over wires” hypothesis</i>	129
5.1	Introduction	129
5.2	Results	130
5.2.1	Use of the mapped K numbers to identify KEGG genomes	135
5.2.2	Identification of protein co-annotation across genomes	136
5.2.3	Method evaluation: determination of expected identified genomes	142
5.2.4	Selecting genome subsets through different filtering methods	143
5.2.5	Generation of testable hypotheses based on literature findings	150
5.3	Discussion	150

5.4	Methods . . . . .	154
5.4.1	Identify genomes from K numbers . . . . .	154
5.4.2	Selecting genome subsets using biochemical process information . . . . .	158
5.4.3	Generation of testable hypotheses based on literature findings . . . . .	160
<b>Chapter 6 Conclusions</b>		<b>161</b>
6.1	Development of an electrochemical platform . . . . .	161
6.2	Investigation of the “syntrophy over wire” hypothesis . . . . .	162
6.3	Development of a computational tool to analyse the relationship between genetic information and biological function . . . . .	163
6.4	Achievements and possible future developments . . . . .	163
<b>References</b>		<b>165</b>
<b>Appendix A Thermodynamics and metabolism</b>		<b>189</b>
A.1	Thermodynamics and metabolism . . . . .	189
<b>Appendix B Electrochemical platform</b>		<b>191</b>
B.1	Electrochemical cell measurements . . . . .	191
B.2	OpenSCAD code for 3D printed objects . . . . .	191
B.2.1	Rubber gasket . . . . .	191
B.2.2	Carbon fibre twill preparation . . . . .	193
B.2.3	Rubber stopper modification . . . . .	193
B.2.4	Electrode support . . . . .	195
B.3	Determining the material used for WE and CE connections . . . . .	196
B.4	Anaerobic conditions . . . . .	197
<b>Appendix C Anaerobic methods and solutions</b>		<b>198</b>
C.1	CCM– related stock solutions . . . . .	198
C.1.1	Trace metal stock solution (100 X) . . . . .	198
C.1.2	Vitamin stock solution (1000 X) . . . . .	198
C.1.3	Cysteine–HCl stock solution (100 X) . . . . .	198
C.1.4	Na <sub>2</sub> S stock solution (50 X) . . . . .	199
C.2	Protocol for CCM preparation . . . . .	199
C.3	Headspace replacement method . . . . .	200
C.4	Anaerobic cryostocks preparation protocol . . . . .	200
C.4.1	Solution preparation . . . . .	200
C.4.1.1	0.1 M phosphate buffer pH 7 (200 mL) . . . . .	200

C.4.2	~100 mM titanium citrate solution (325 mL)	200
C.4.2.1	50% Glycerol solution (v/v)	200
C.4.3	Procedure	200
<b>Appendix D</b>	<b>Electrochemical experiment</b>	<b>201</b>
D.1	Current measurements	201
D.1.1	Estimation of the number of electrons exchanged	201
D.2	EIS	204
D.3	CV	206
D.4	Ion chromatography	211
D.4.1	Standard curve and calculation of the sample concentrations	211
D.4.2	Change in compound concentration for the samples	212
D.4.3	Statistical analyses of the change of sample concentrations	212
D.4.3.1	Lactate	212
D.4.3.2	Acetate	213
D.4.4	Growth analyses of the control cultures	216
<b>Appendix E</b>	<b>MetQy</b>	<b>219</b>
E.1	MetQy publication	220
E.2	MetQy package documentation	224
<b>Appendix F</b>	<b>Bioinformatics analysis using MetQy</b>	<b>272</b>
F.1	Methods	272
F.1.1	Hack to obtain the <code>genome_reference_table</code> object	272

# List of Tables

2.1	MUX channels connections . . . . .	40
2.2	Specification of electrochemical cell components . . . . .	41
2.3	Specification of the reference electrode components . . . . .	44
2.4	Specification of the electronic components . . . . .	50
2.5	Electronic connections from Gamry to electrochemical cell electrodes for ZRA mea- surements . . . . .	52
3.1	Summary of Dv's enzymes potentially involved in electron transfer. . . . .	54
3.2	Summary of Mm's enzymes potentially involved in electron transfer. . . . .	55
3.3	EIS model parameter comparison: across conditions and time points . . . . .	61
3.4	Estimated mean potential ( $E_p$ ) and current ( $I_p$ ) peaks . . . . .	65
3.5	Change in acetate concentration across conditions: post hoc output . . . . .	68
3.6	Experimental design summary . . . . .	77
3.7	Control culture tubes summary . . . . .	78
3.8	Electrochemical cell arrangement within containers . . . . .	82
3.9	Electronic connections for ZRA measurements . . . . .	83
3.10	Electronic connections for EIS measurements . . . . .	86
3.11	Electronic connections for CV measurements . . . . .	87
4.1	KEGG databases in MetQy . . . . .	94
4.2	Summary of KEGG genome data in MetQy . . . . .	96
5.1	Mapping of Dv's genes potentially involved in electron transfer to K numbers. . . .	131
5.2	Mapping of Mm's genes potentially involved in electron transfer to K numbers. . .	132
5.3	Summary of KEGG genomes identified using the anodic protein search. . . . .	135
5.4	Summary of KEGG genomes identified using the cathodic protein search. . . . .	135
5.5	Co-occurrence frequency of genomes annotated with the anodic proteins. . . . .	136
5.6	Co-occurrence frequency of genomes annotated with the cathodic proteins. . . . .	136
5.7	Protein co-occurrence for each phylum based on Dv's protein search. . . . .	138



5.8	Protein co-occurrence for each phylum based on Mm's protein search. . . . .	139
5.9	Protein co-occurrence of the genus <i>Desulfovibrio</i> found when searching for Mm's proteins . . . . .	142
5.10	Protein co-occurrence of known electroactive genera. . . . .	142
5.11	Electroactive organisms identified by both the anodic and cathodic protein searches.	143
5.12	Workable experiment organisms based on the anodic and cathodic protein searches.	144
5.13	Cyanobacteria identified using the anodic protein search. . . . .	145
5.14	Cyanobacteria identified using the cathodic protein search. . . . .	146
5.15	Genomes found to be annotated with Hyn. . . . .	149
5.16	Genomes found to be annotated with Eha, Ehb and Frc/Fru. . . . .	149
5.17	Co-occurrence of cathodic genomes annotated with Eha, Ehb and Frc/Fru. . . . .	150
A.1	Reduction potential chart – a thermodynamic view of metabolic redox reactions. .	189
A.2	Microorganisms viewed from a thermodynamic perspective . . . . .	190
C.1	Trace metal stock solution (100 X, 1 L) . . . . .	198
C.2	Vitamin stock solution (1000 X, 1 L) . . . . .	199
D.1	Number of Coulombs estimated . . . . .	201

# List of Figures

1.1	Reduction potential chart – an explanation . . . . .	4
1.2	Reduction potential chart – a thermodynamic view of metabolic redox reactions. .	5
1.3	Microorganisms viewed from a thermodynamic perspective . . . . .	6
1.4	Emerging constraints due to the reduction potential of metabolic half reactions . .	7
1.5	Overview of metabolic interaction motifs between two species . . . . .	9
1.6	Bioelectrochemical systems . . . . .	14
1.7	Dv – Mm coculture . . . . .	17
2.1	Complete system . . . . .	21
2.2	Electrochemical cell schematic . . . . .	23
2.3	CFTS-based electrodes . . . . .	24
2.4	Reference electrode production: method 1 . . . . .	25
2.5	Reduction potential translation across reference systems . . . . .	26
2.6	Inside view of the electrochemical cell . . . . .	27
2.7	Production of the <i>modified stopper</i> . . . . .	28
2.8	Assembled electrochemical cell . . . . .	29
2.9	Container schematic – holes . . . . .	30
2.10	Container wall connectors . . . . .	31
2.11	Container schematic – wiring . . . . .	32
2.12	Connections . . . . .	33
2.13	Assembled container . . . . .	34
2.14	Picture of the complete system . . . . .	35
2.15	Rubber gasket preparation . . . . .	42
2.16	Working and counter electrode manufacturing . . . . .	43
2.17	3D printed electrode support . . . . .	44
2.18	Set up required to characterise the reference electrodes . . . . .	46
2.19	Electrochemical cell assembly . . . . .	49
2.20	Assembly of the internal connection components . . . . .	51

3.1	Potential mechanisms involved in the interaction between Dv and Mm and the electrodes . . . . .	57
3.2	Current . . . . .	59
3.3	Estimation of the number of electrons exchanged . . . . .	60
3.4	EIS bode before and after current monitoring . . . . .	62
3.5	EIS Nyquist before and after current monitoring . . . . .	63
3.6	EIS – Boxplot of fitted parameter values of Randles cell model . . . . .	64
3.7	CV traces . . . . .	65
3.8	Boxplot of the peak potentials . . . . .	66
3.9	Change in compound concentration across conditions . . . . .	67
3.10	Change in compound concentration by treatment type. . . . .	68
3.11	Growth curves of control cultures . . . . .	70
3.12	Comparison of control tube growth with change in compound concentration . . . .	71
3.13	EIS model . . . . .	85
3.14	Method to achieve comparable CV . . . . .	87
3.15	Gradient used for ion separation. . . . .	90
4.1	Example of plot generated by <i>plot_heatmap</i> . . . . .	109
4.2	Example of plot generated by <i>plot_scatter</i> . . . . .	110
4.3	Example of plot generated by <i>plot_scatter_byFactors</i> . . . . .	111
4.4	Example of plot generated by <i>plot_sunburst</i> . . . . .	113
4.5	Example of plot generated by <i>plot_variance_boxplot</i> . . . . .	114
4.6	Biological example: Step 2 figure . . . . .	117
4.7	Biological example: Step 5 figure . . . . .	119
4.8	Biological example: Step 6.2 . . . . .	122
4.9	Biological example: Step 6.3 . . . . .	122
4.10	Biological example: Step 6.4 . . . . .	123
4.11	Biological example: Step 6.6 – mean <i>mcf</i> by genus . . . . .	123
4.12	Biological example: Step 6.6 – <i>mcf</i> standard deviation by genus . . . . .	124
4.13	Biological example: Step 6.7 figures . . . . .	124
4.14	Biological example: Steps 6.8 – PC plot . . . . .	125
4.15	Biological example: Steps 6.8 – mean Euclidean distance plot . . . . .	125
5.1	PC plot of all the cathodic organisms identified . . . . .	147
5.2	PC plot of all the anodic organisms identified . . . . .	148
B.1	Electrochemical cell measurements . . . . .	191

B.2	Electrode wire composition determination . . . . .	197
B.3	Container – hole mask . . . . .	197
D.1	Current over time for the three periods monitored . . . . .	202
D.2	Voltage recorded over time during current measurement . . . . .	203
D.3	EIS nyquist . . . . .	204
D.4	OCP before and after EIS . . . . .	205
D.5	CV peak determination: ranges used . . . . .	206
D.6	CV peak determination: CV trace and first derivative (slopes) . . . . .	207
D.7	CV peak determination: scaled slope minus the current in range . . . . .	207
D.8	CV of abiotic (A) electrochemical cells . . . . .	208
D.9	CV of biotic (B) electrochemical cells . . . . .	209
D.10	CV of non-connected biotic (nB) electrochemical cells . . . . .	210
D.11	Standard curve and calculation of the sample concentrations . . . . .	211
D.12	Change in compound concentration for the samples . . . . .	212
D.13	Growth curves of control cultures . . . . .	217
D.14	Growth curves of culture tubes at 37 °C . . . . .	218

# List of Code

3.1	Peak identification . . . . .	88
4.1	MetQy installation steps . . . . .	98
4.2	Usage example for <i>parseKEGG_compound</i> . . . . .	100
4.3	Usage example for <i>parseKEGG_ko_enzyme</i> . . . . .	100
4.4	Usage example for <i>query_genomes_to_modules</i> – input: organisms’ names . . . . .	101
4.5	Usage example for <i>query_genomes_to_modules</i> – input: gene sets . . . . .	101
4.6	Usage example for <i>query_genomes_to_modules</i> – input: T numbers . . . . .	102
4.7	Usage example for <i>query_genomes_to_modules</i> – input: gene sets . . . . .	102
4.8	Usage example for <i>query_genomes_to_modules</i> – input: T numbers . . . . .	102
4.9	Usage example for <i>query_genomes_to_modules</i> – output details . . . . .	103
4.10	Usage example for <i>query_genomes_to_modules</i> – input: T numbers . . . . .	103
4.11	Usage example for <i>query_modules_to_genomes</i> . . . . .	103
4.12	Usage example for <i>query_genes_to_modules</i> . . . . .	104
4.13	Usage example for <i>query_genes_to_genomes</i> . . . . .	104
4.14	Usage example for <i>query_missingGenes_from_module</i> . . . . .	105
4.15	Usage example for <i>analysis_pca_mean_distance_calculation</i> . . . . .	107
4.16	Usage example for <i>analysis_pca_mean_distance_grouping</i> . . . . .	107
4.17	Usage example for <i>plot_heatmap</i> . . . . .	109
4.18	Usage example for <i>plot_heatmap</i> – ordered data . . . . .	109
4.19	Usage example for <i>plot_scatter</i> – plot group values . . . . .	110
4.20	Usage example for <i>plot_scatter</i> – plot group values coloured by the ‘ <i>FACTOR</i> ’ . . .	110
4.21	Usage example for <i>plot_scatter</i> – plot group values in the order given . . . . .	110
4.22	Usage example for <i>plot_scatter_byFactors</i> . . . . .	111
4.23	Usage example for <i>plot_scatter_byFactors</i> . . . . .	112
4.24	Usage example to generate the data for <i>plot_sunburst</i> . . . . .	112
4.25	Usage example for <i>plot_sunburst</i> . . . . .	112
4.26	Usage example for <i>plot_sunburst</i> – colouring the outer ring . . . . .	113
4.27	Usage example for <i>plot_sunburst</i> – colouring the outer and inner rings . . . . .	113

4.28	Usage example for <i>plot_variance_boxplot</i> . . . . .	114
4.29	Biological example: step 1 . . . . .	115
4.30	Biological example: step 2 . . . . .	117
4.31	Biological example: step 3 . . . . .	118
4.32	Biological example: step 4 . . . . .	119
4.33	Biological example: step 5 . . . . .	120
4.34	Biological example: step 6 . . . . .	121
5.1	Identifying genomes from K numbers . . . . .	154
5.2	Use of <b>MetQy</b> to determine organism subset . . . . .	158
5.3	Using <b>MetQy</b> to check the possible pairing of a known coculture in the context of the “syntrophy over wires” hypothesis . . . . .	160
B.1	Rubber gasket preparation – 3D printed stencil . . . . .	192
B.2	3D printed electrode stencil – frame . . . . .	193
B.3	Rubber bun preparation – 3D printed holder . . . . .	193
B.4	Rubber bun preparation – 3D printed holder support . . . . .	194
B.5	3D printed electrode support . . . . .	195
F.1	Retrieving genome information contained in <b>genome_reference_table</b> object using <b>MetQy</b> functions . . . . .	272

# Acknowledgments

I would like to thank those that made this experience that much better in so many different ways (in no particular order).

My family, especially my parents, as I would not be here if it were not for their love and support and making sure I had a great education.

My supervisor, Orkun Soyer. Although challenges were met, I learnt a great deal from you and I appreciate your kindness and patience.

Kevin Purdy, thank you for taking on a pseudo supervisory role and making the process easier. Words fall short of the extent of my thanks and appreciation.

Kalesh, thank you for willing taking on the role of mentor, for both life and academic matters. Thank you for teaching so much about so many things and for always helping and supporting me any any and every way you could. And also thank you for believing in me and making sure I did in moments of doubt.

Sean, it feels you were a partner in crime and time and I would not have survived without you there for endless cups of coffee and all sorts of conversations. The skin robot shall for ever be remembered!

Daniel Franklin, thank you for all your mentoring and encouragement with my teaching and generally lending a listening ear or reading eye when needed.

Christian and James Stratford, who helped and supported me throughout my PhD and eased me into the world of electrochemistry.

To all past and present OSS lab members, thank you! Particularly, Kelsey, thank you for your help

in the end! Jing, one could not ask for a better lab mate. Thank you for all your help throughout the years. Fred, thank you for teaching me about logic evaluation of expressions in programming.

To all present and past inhabitants of Greenwood Court, thank you for all the delicious dinners, movie nights and support. Special thanks to Aurelija and Elena, two of my best friends, who have made sure I stayed fed and sane throughout. Also thanks to Clare for introducing me into the world of “camp” movies and also helping proof-read a chapter.

Ale, thank you for everything. I wouldn’t have survived without your support and words of wisdom.

Thank you to all my friends in Mexico and beyond who have stuck by me and lent your support, specially Auryn, Martin and Tania.

To anyone who feels I should have mentioned them but didn’t, I am sure you’re absolutely right and I thank you too as well!

This thesis was typeset with  $\text{\LaTeX} 2_{\epsilon}$ <sup>1</sup> by the author.

---

<sup>1</sup>  $\text{\LaTeX} 2_{\epsilon}$  is an extension of  $\text{\LaTeX}$ .  $\text{\LaTeX}$  is a collection of macros for  $\text{\TeX}$ .  $\text{\TeX}$  is a trademark of the American Mathematical Society. The style package *warwickthesis* was used.



# Declarations

I hereby declare that this dissertation entitled “Tool development for metabolic analyses in the context of thermodynamic constraints” is an original work and has not been submitted for a degree or diploma or other qualification at any other University.

Chapter 1 introduces the background and aims for this study.

Chapter 2 explains the electrochemical platform I developed with contributions listed below. I performed or conceptualised any aspect of the chapter not listed below.

- Electrochemical cell design – Daniel Carlotta–Jones<sup>1</sup> and Dr. James Stratford<sup>2,3</sup>
- *Modified stopper*

Design – Dr. James Stratford and Andrea S. Martinez–Vernon

Initial conceptual design of 3D printed stencils – Dr. Kalesh Sasidharan<sup>2</sup> and Andrea S. Martinez–Vernon<sup>2,4</sup>

Computer aided design (CAD) – Andrea S. Martinez–Vernon

- Concept of utilising a container with connectors through its wall – Dr. James Stratford<sup>2,3</sup>

Hole and connector design – Andrea S. Martinez–Vernon

Note that I performed all of the work included in this chapter and the collaborations were established at the conceptual level.

---

<sup>1</sup>Warwick Manufacturing Group (WMG), University of Warwick

<sup>2</sup>School of Life Sciences, University of Warwick

<sup>3</sup>Warwick Integrative Synthetic Biology Centre (WISB), University of Warwick

<sup>4</sup>BBSRC/EPSRC Synthetic Biology Research Centre, Universities of Bristol, Oxford and Warwick

Chapter 3 described the implementation of the platform described in Chapter 2 to test an original hypothesis (refer to Section 1.7.1). I made Figure D.14 (in this chapter’s appendix) with unpublished data generated by Dr. Jing Chen<sup>2</sup> in 2015.

Chapter 4 is based on the published paper for which I am first author (Martinez-Vernon et al. 2018) listed in the Publications section below.

The work relevant for the second paper (Sasidharan et al. 2018) has not been included in this thesis, but its application in the platform described in Chapter 2 is discussed in its Discussion section.

## Publications

- Martinez-Vernon, A. S., Farrell, F., & Soyer, O. S. (2018). MetQy - an R package to query metabolic functions of genes and genomes. *Bioinformatics* 34(23).  
<https://doi.org/10.1093/bioinformatics/bty447>
- Sasidharan, K., Martinez-Vernon, A.S., Chen, J., Fu, T., & Soyer, O. (2018). A low-cost DIY device for high resolution, continuous measurement of microbial growth dynamics. *BioRxiv*, 407742.  
<https://doi.org/10.1101/407742>

## Funding

This work was funded by The University of Warwick and by the Biotechnological and Biological and Engineering and Physical Sciences Research Councils (BB- and EPSRC), with grant IDs: EP/L016494/1 (to the Centre for Doctoral Training in Synthetic Biology, SynBioCDT), BB/K003240/2 (to OSS), BB/M017982/1 (to the Warwick Integrative Synthetic Biology Centre, WISB).

---

<sup>1</sup>Warwick Manufacturing Group (WMG), University of Warwick

<sup>2</sup>School of Life Sciences, University of Warwick

<sup>3</sup>Warwick Integrative Synthetic Biology Centre (WISB), University of Warwick

<sup>4</sup>BBSRC/EPSRC Synthetic Biology Research Centre, Universities of Bristol, Oxford and Warwick

# Abstract

Metabolism is key to all biological processes. Studies have yet to establish the link between metabolism, extracellular electron transfer and thermodynamics. In this context, I successfully developed a platform to enable electrochemical experiments using strict anaerobic microorganisms to quantify the electron transfer in an effort to measure metabolic rates. This involved establishing a novel hypothesis to investigate “syntrophy over wires”. Moreover, I developed a computational tool, **MetQy**, to enable the automated, large-scale analysis of annotated genomes with metabolic information in the form of an R package. The work I presented here has paved the way for electrochemical and computational analyses towards characterising and better understanding metabolic–electronic interactions in the context of thermodynamics.

# Abbreviations

ATP	Adenosine triphosphate
AUX	Auxiliary electrode
BES	Bioelectrochemical system
BP	Banana plug
BS	Banana socket
C	Counter connection in Gamry potentiostat 600+
CE	Counter electrode
CFT	Carbon fibre twill
CFTS	Carbon fibre twill square
CS	Counter sense connection in Gamry potentiostat 600+
CV	Cyclic voltammetry
dia.	Diameter
Dv	<i>Desulfovibrio vulgaris</i>
DvMm	Coculture between Dv and Mm
EET	Extracellular electron transfer
EIS	Electrochemical Impedance Spectroscopy
ext.	External
IET	Interspecies electron transfer
IHT	Interspecies hydrogen transfer
KEGG	The Kyoto Encyclopedia of Genes and Genomes

KFeCN Potassium ferricyanide

*mcf module completeness fraction*

MEC Microbial electrolysis cell

MESC Microbial electrosynthesis cell

MFC Microbial fuel cell

Mm *Methanococcus maripaludis*

MUX Multiplexer (Multichannel potentiostat associated with GAMRY potentiostat)

OCP Open circuit potential

PC(s) Principal component(s)

PCA Principal component analysis

PLA Polylactic acid

PTFE Polytetrafluoroethylene

Ref Reference electrode

SHE Standard hydrogen electrode

SS Stainless steel

W Working connection in Gamry potentiostat 600+

W1 Working electrode 1

W2 Working electrode 2

WE(s) Working electrode(s)

WS Working sense connection in Gamry potentiostat 600+

RE Reference electrode

REF Reference connection in Gamry potentiostat 600+

TEA Terminal electron acceptor

WW Waste water

ZRA Zero resistance ammeter

# Chapter 1

## Introduction

Metabolism is key to all biological processes as it comprises a series of chemical reactions that enable the oxidation of chemical compounds with the ultimate aim of harvesting the energy required for compound synthesis, mechanical work and all other cell functions, as well as the molecules required for cellular function and maintenance (Alberts et al., 2002). Catabolism is the subset of metabolic reactions which involve enzymes catalysing the breakdown of complex organic molecules to simpler ones in multiple steps releasing energy. The energy is temporally stored in small diffusible molecules, such as NADH, called electron carriers. The molecules formed in the intermediary steps during catabolism are called catabolites. Anabolism, or biosynthesis, is the subset of metabolic reactions which makes use of the released energy and smaller molecules to achieve cell growth and maintenance (Alberts et al., 2002). Therefore, metabolism underpins most of cellular physiology and its capacity to produce a whole range of chemical structures has been taken advantage in biotechnological applications, which refer to the use of organisms to make useful products.<sup>1</sup>

### 1.1 Thermodynamics

Metabolic reactions, like all chemical reactions, have energetic considerations attached to them, which have been largely ignored in the study of cellular metabolism in the genomic era (Soh and Hatzimanikatis, 2010). The metabolism of a particular organism can be studied as a system with thermodynamic properties. A central thermodynamic property is Gibbs free energy ( $G$ ), which is a measure of the total energy to do work in that system. For a system state to change spontaneously, a negative change in Gibbs free energy,  $-\Delta G$ , is necessary, meaning that energy is being released (exergonic process).  $+\Delta G$  would indicate that energy was needed to carry out the reaction (endergonic process) and, hence, the system would suffer from thermodynamic inhibition

---

<sup>1</sup><https://archive.is/20130414170840/http://www.europabio.org/what-biotechnology>

(Alberty, 2003; Price et al., 2001; Schink, 1997; Großkopf and Soyer, 2016).

The standard reaction Gibbs free energy ( $\Delta_r G^o$ ) is calculated by adding the Gibbs free energy of formation ( $\Delta_f G_i^o$ ) of each of the species involved in the reaction under standard conditions (1 M, pH 0, 25°C, 1 atm) (Equation 1.1)

$$\Delta_r G^o = \sum_{i=1}^{N_S} v_i \Delta_f G_i^o \quad (1.1)$$

where  $N_S$  is total number of chemical species and stoichiometric coefficient ( $v_i$ ) is positive for products and negative for reactants. The standard Gibbs energy of formation ( $\Delta_f G_i^o$ ) of species  $i$  is the Gibbs energy change when a mole of the species in its standard state (1 bar or 1 M) is formed from its elements in their reference states. When reactants and products are not found at standard pressure or concentration, it is possible to calculate the reaction Gibbs free energy ( $\Delta_r G$ ) (Equation 1.2)

$$\Delta_r G = \Delta_r G^o + RT \ln \left( \prod_{i=1}^{N_S} c_i^{v_i} \right) \quad (1.2)$$

where  $c_i$  is the non-standard concentrations of the chemical species,  $T$  is the temperature in Kelvin and  $R$  is the gas constant ( $8.314 J mol^{-1} K^{-1}$ ) (Alberty, 2003).

When dealing with biochemical systems, it is preferable to consider the bioenergetics that occur under physiological conditions rather than at pH 0. Therefore, some work has been done (e.g. Alberty, 2001; Thauer et al., 1977) to establish the Gibbs energy for biochemical reactions under standard biochemical conditions ( $\Delta_r G^{o'}$ , 1 M, pH 7, 25°C, 1 atm), where all the concentrations are still kept at 1 M, except for that of  $[H^+]$ , which is kept at  $10^{-7}$  M (pH 7) and is denoted by a prime ( $'$ ) (Alberty, 2003).  $\Delta_r G^{o'}$  and  $\Delta_f G_i^{o'}$  would be substituted into Equations 1.1 and 1.2 when describing systems under standard biochemical conditions.

Reduction–oxidation (redox) reactions involve the transferring of electrons from one chemical species to another. These reactions can be described as an oxidation half reaction, consisting of the chemical species that loses electrons (electron donor), and a reduction half reaction, consisting of the chemical species that gains electrons (electron acceptor). An oxidation half reaction written in the opposite or reverse direction would be a reduction half reaction. Each half reaction has a reduction potential that can be measured using an electrode as the electron source or sink, using a reference electrode. Here, all reduction potential values are referred to against the Standard Hydrogen Electrode (SHE) as a reference. For a redox reaction to occur, it must be complete (i.e. there must be one chemical species being oxidised and another being reduced) and spontaneous (the change in Gibbs free energy must be negative or the reduction potential of the reaction must be positive). The potential difference of a reaction ( $E_h^{o'}$ ) can be directly calculated from the change

in Gibbs free energy (Equation 1.3)

$$\Delta E_h^{o'} = -\frac{\Delta G^{o'}}{z \cdot F} \quad (1.3)$$

where  $F$  is the Faraday constant (96.4853 kJ per volt gram equivalent),  $z$  is the number of electrons ( $e^-$ ) transferred in the reaction and  $E_h^{o'}$  is the potential difference against SHE (v SHE) under standard biochemical conditions. The advantage of using reaction energetics with the  $E_h^{o'}$  is that the value was normalised by the number of electrons exchanged ( $z$ ) and can, therefore, be directly compared between redox reactions.

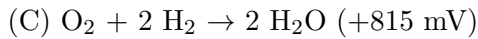
Alternatively, the potential difference of a redox reaction can be calculated by adding the potential of each half reaction (Equation 1.4)

$$E_h^{o'} = E_{reduction}^o + E_{oxidation}^o \quad (1.4)$$

where  $E_{reduction}^o$  and  $E_{oxidation}^o$  refer to the potential of reduction and oxidation half reaction, respectively. Since the oxidation half reaction can be described as a reduction half reaction in the reverse direction,  $E_{oxidation}^o$  can be written as  $-E_{reduction}^{o'}$ , where the prime ( $'$ ) denotes it has been reversed. Equation 1.4 can be rewritten as Equation 1.5:

$$E_h^{o'} = E_{reduction}^o - E_{reduction}^{o'} \quad (1.5)$$

As mentioned before, for a reaction to occur spontaneously, it must have a negative  $\Delta_r G$  or a positive  $E_h^{o'}$ . Otherwise, the reaction in question is termed endergonic, as it requires the addition of energy for it to take place (Alberty, 2003). To illustrate this concept, a worded and visual example have been included hereunder (Figure 1.1). Suppose there are three reduction half reactions (listed below) with their corresponding reduction potential ( $E_{reduction}^o$ ):



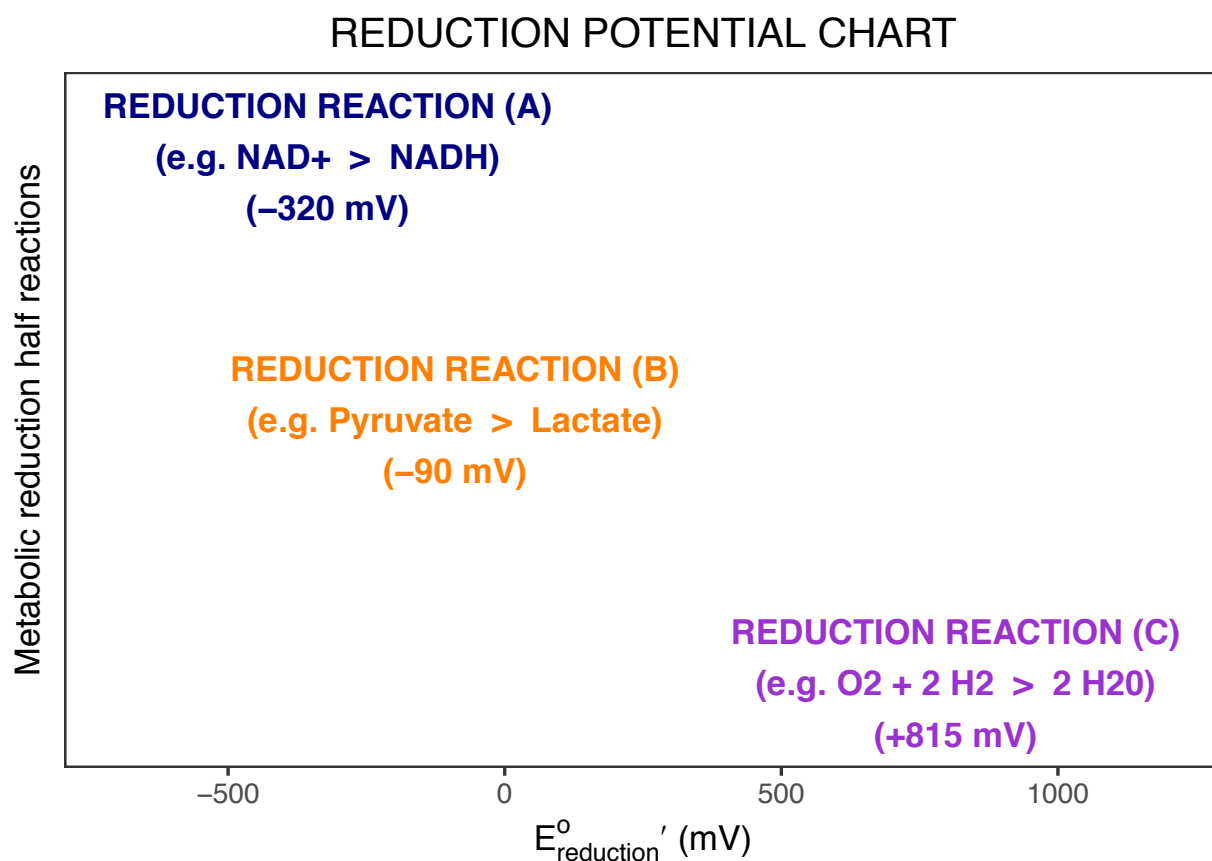
Throughout this work, all half reactions have been represented as reduction half reactions to make their representation and comparison easier. In order to carry out the reduction of pyruvate to lactate (B), this reduction half reaction needs to be coupled with reduction half reaction (A) in the reverse direction, i.e. (A')  $\text{NADH} \rightarrow \text{NAD}^+$ . Using Equation 1.5, the potential for this redox reaction ( $E_h^{o'}$ ) would be  $-190\text{mV} - (-320\text{mV}) = -190 + 320 = 130 \text{ mV}$ . Since  $E_h^{o'} > 0$ , the reaction is feasible. However, the reduction of pyruvate could not be coupled with the oxidation of  $\text{H}_2\text{O}$  (reduction reaction (C)), as the redox reaction potential would be  $-190\text{mV} - (+815\text{mV}) = -190 - 815 = -1005 \text{ mV}$ , a thermodynamically infeasible reaction.



## 1.2 Metabolism and thermodynamics

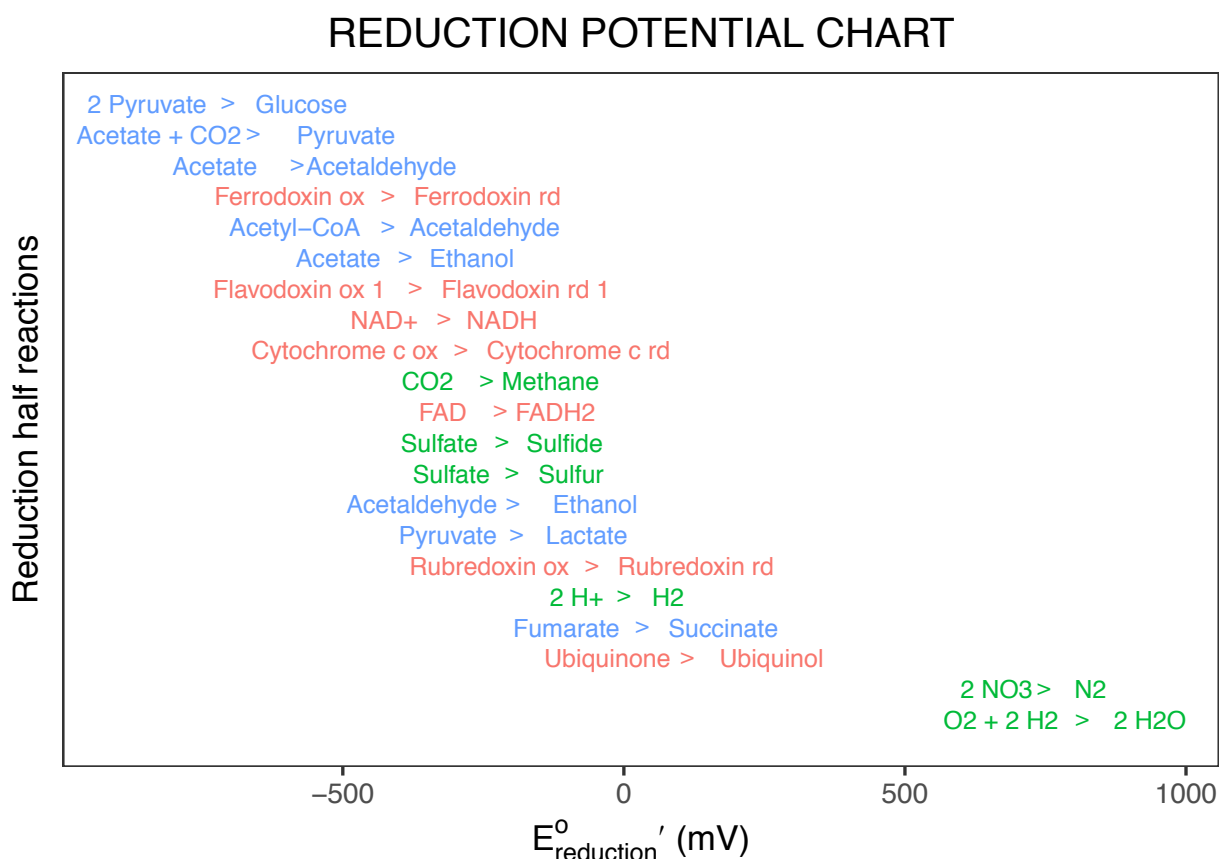
Metabolism comprises a series of chemical reactions that enable the oxidation of organic compounds with the ultimate aim of harvesting energy (Alberty, 2003), as mentioned before. Energy is harvested by oxidising a high-energy molecule (e.g. glucose) down to small low-energy reduced molecules (e.g.  $\text{CH}_4$ ), which is a thermodynamic process. The stepwise reduction of the total potential energy through a chain of redox systems is made possible by a number of coupled reactions that result in the production, conservation and release of energy. The cell is able to conserve part of the energy as chemical energy (ATP). This stepwise electron carrier system ends with the respiratory chain, which allows electrons to be transferred to a terminal electron acceptor (TEAs) (Doelle, 1975). When microorganisms use external TEAs, the metabolism is called respiration, while the metabolism that uses internal TEAs is called fermentation (Rabaey, 2010).

The redox reactions in metabolism can be used to calculate the energy gain possible through



**Figure 1.1** Reduction potential chart – an explanation. The x-axis shows the reduction potential, while the y-axis lists the reduction half reactions in a readable fashion. The example reduction half reactions with their respective reduction potential are (A)  $\text{NAD}^+ \rightarrow \text{NADH}$  (-320 mV), (B)  $\text{Pyruvate} \rightarrow \text{Lactate}$  (-190 mV) and (C)  $\text{O}_2 + 2 \text{H}_2 \rightarrow 2 \text{H}_2\text{O}$  (+815 mV). The reduction half reaction (B) can be coupled with reduction half reaction (A) in the reverse direction (with a reduction potential of +320 mV), leading to a potential of the redox reaction of 130 mV. However, it could not be coupled with (C) as this would result in a positive potential for the redox reaction ( $-190 - 815 = -1005$  mV; i.e. infeasible). Data obtained from Thauer et al. (1977) and Alberty (2001). Note that the half reaction labels are centred on their respective reduction potential values.

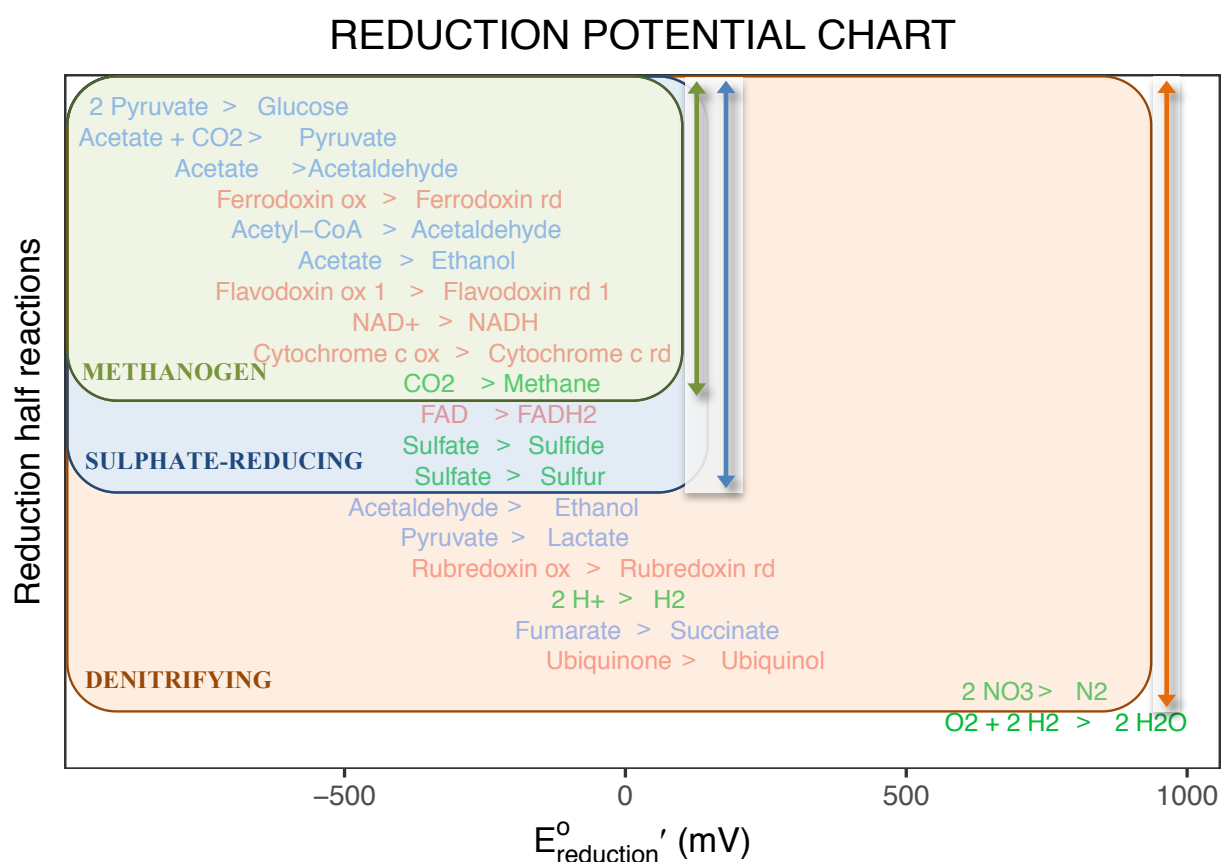
the oxidation of a substrate and the reduction of a TEA by the difference in their reduction potential (Rabaey, 2010). Therefore, the metabolic redox reactions, which can be written as half reactions, as described in Section 1.1, can be represented in a reduction potential chart similar to the one used in Figure 1.1. Figure 1.2 (see Table A.1 for a tabular representation) shows some examples of half reactions that occur in metabolism written as reduction half reactions. These are represented in the reduction potential chart and Figure 1.1 was used to explain how the potential of a redox reaction can be calculated from the potential of the two half reactions involved. The reduction half reactions in Figure 1.2 can be distinguished by the type of molecules involved, such as organic compounds (blue), electron carriers (red) or TEAs (green). Electron carriers (also called electron shuttles or redox mediators) are organic molecules that can be reversibly oxidised (ox) and reduced (rd) and can, therefore, mediate the transfer of electrons in biological systems (i.e. cells) (Roehm, 2001; Watanabe et al., 2009). The organic compounds are usually oxidised to



**Figure 1.2** Reduction potential chart – a thermodynamic view of metabolic redox reactions. The reduction potential of some reduction half reactions that occur in microbial metabolism are plotted along the x-axis based on their reduction potential ( $E^{\circ}_h$ ). For simplicity, some multi-step reactions in metabolism, such as pyruvate to glucose, are represented as a single step reaction. In these cases, the reduction potential of the overall reaction is given (system property, Alberty, 2003). The y-axis only lists the reactions in order to separate them. The half reactions have been colour-coded depending on whether they involve organic compounds (blue), electron carriers (red) or TEAs (green). Data obtained from Thauer et al. (1977) and Alberty (2001). Note that the half reaction arrows (>) are positioned at their respective reduction potential values. ox, oxidised; rd, reduced; FAD, flavin adenine dinucleotide.

harvest energy and building blocks, but they can also be used as TEA (called fermentation). It can be observed that the TEA with the highest potential is  $O_2$ . This means that organisms that use  $O_2$  as a TEA are able to extract more energy than other organism using a different TEA when oxidising the same substrate (Rabaey, 2010).

After mapping those metabolic half reactions on the reduction potential scale, we can describe microorganisms as metabolic entities with a ‘redox profile’. The ‘redox profile’ can be defined as the set of half reactions that each organism can carry out. Figure 1.3 (see Table A.2 for a tabular representation) represents the expected profile of three different groups of microorganisms characterised by the biochemical processes that they perform. Methanogens and sulphate-reducing and denitrifying bacteria produce methane ( $CH_4$ ) by reducing  $CO_2$ , reduce sulphate and remove

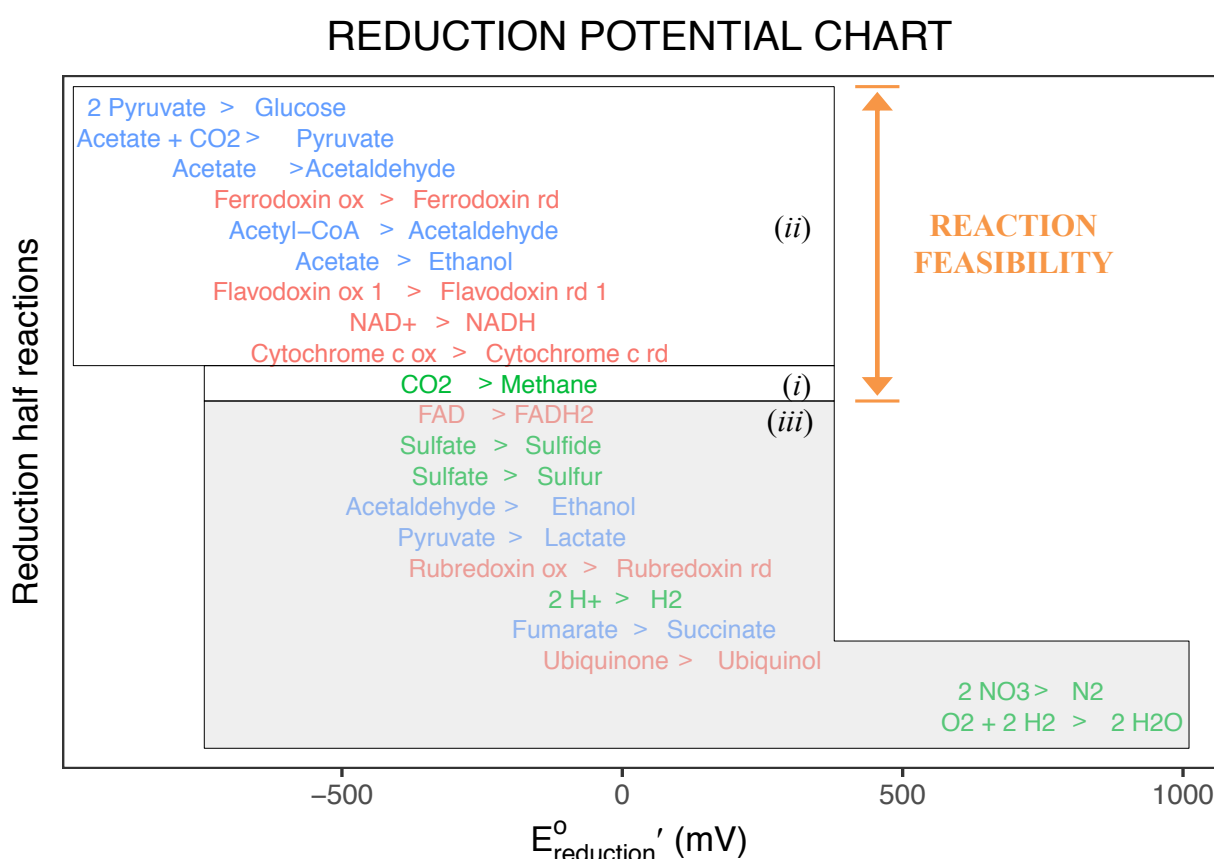


**Figure 1.3** Three types of microorganisms were roughly mapped onto the reduction potential chart based on known metabolic processes they carry out, each represented by circles of different colour. Methanogens (green) are known to reduce  $CO_2$  to  $CH_4$ . Sulphate-reducing bacteria (blue) are known to reduce sulphate to either sulphide or sulphur, as their name suggests. Finally, denitrifying bacteria reduce  $NO_3$  to  $N_2$ . These processes refer to the “last” reactions that the organisms can carry out and hence have determined the bottom range for the species assignment. However, the question of the upper bound that should be assigned to each remains open, as it would depend on their genetic capacity. Here, I have assumed they can all oxidise glucose to pyruvate and the arrow show the range of possible half reactions for each species, limiting the maximum energy (in the form of potential difference) available to them. Therefore, methanogens can harvest much less energy than sulphate-reducing or denitrifying bacteria. Data obtained from Thauer et al. (1977) and Alberty (2001). Note that the half reaction arrows ( $>$ ) are positioned at their respective reduction potential values.

nitrate ( $\text{NO}_3$ ), respectively, as their name suggests. Therefore, the biochemical processes by which they are known refer to the TEA used. In these cases, the ‘lower bound’ of their redox profile is easily identified as it is defined by the TEA that the organism is known to use and it has the highest (most positive) reduction potential. The ‘upper bound’ is defined by the carbon source (organic compound) that the microorganisms oxidise, which is more elusive, and it has the lowest (most negative) reduction potential. The difference of these refers to the overall potential energy available to the microorganism (Equation 1.5).

Any thermodynamic system has constraints, as mentioned in Section 1.1. The arrows in Table A.2 highlight these in the context of microbial metabolism. For instance, a methanogen, which uses  $\text{CO}_2$  as its TEA (blue), can only be coupled with the half reactions above it as oxidation half reactions (i.e. in their reverse direction). The half reactions below are hence unavailable. This limits the number of possible half reaction couplings that a microorganism can carry out and thus imposes energetic constraints on living systems.

The main factors that impact the constraint that the pairing of half reactions imposes on



**Figure 1.4** Emerging constraints due to the reduction potential of metabolic half reactions. To illustrate the constraints discussed in the main text, a visual example has been developed. In order to reduce, for instance,  $\text{CO}_2$  to  $\text{CH}_4$  (i), one of the reactions listed above (ii) must be run in the inverse order (as an oxidation reaction) for the  $\text{CO}_2$  reduction to be thermodynamically feasible (see Figure 1.1). This limits the available reactions to be carried out, as depicted by the greyed-out box (iii).

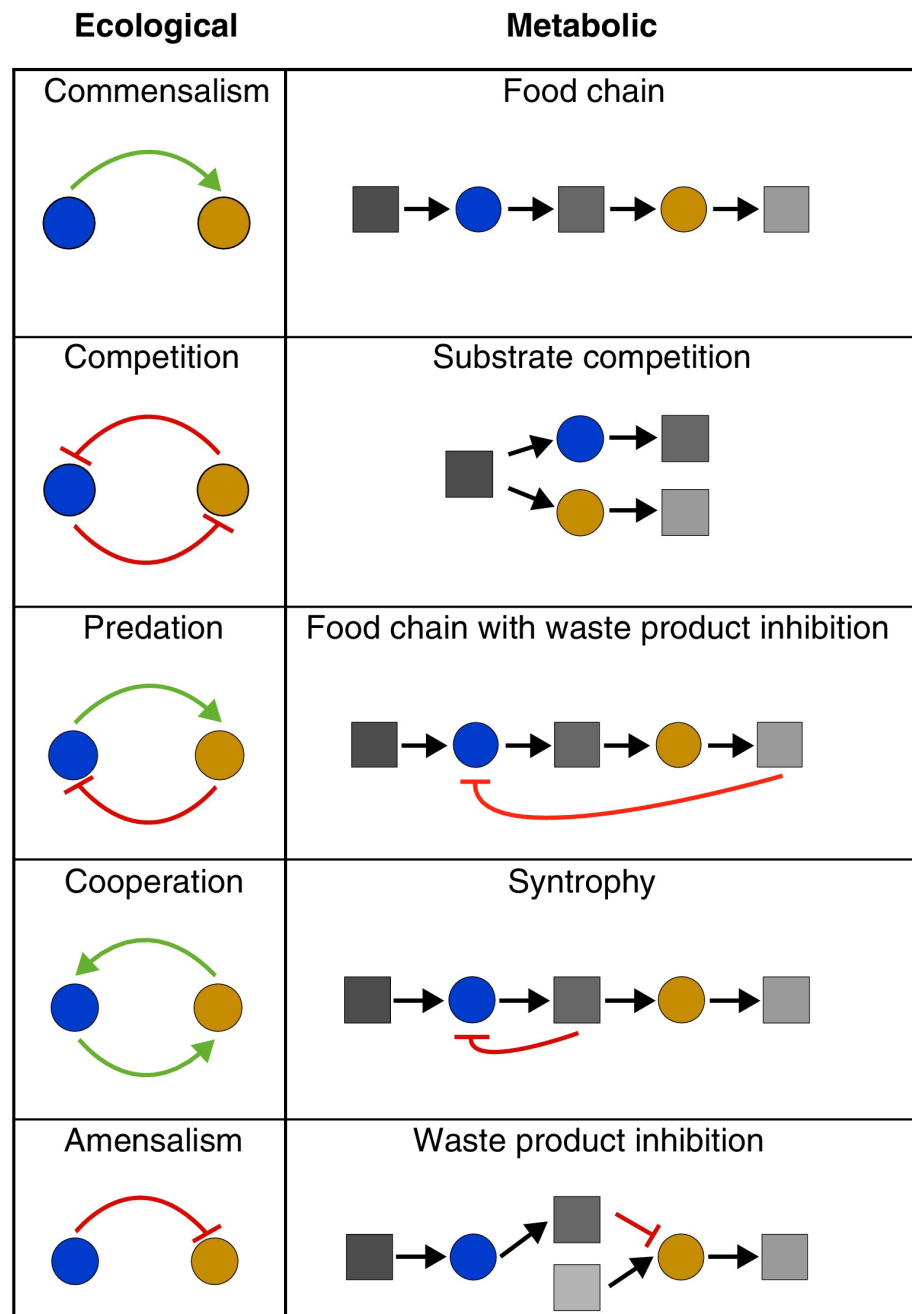
living systems are the environment (e.g. availability of compounds, pH, temperature), the chemical and energetic properties of the compounds being utilised and the microorganisms' ability to oxidise them, given by their enzymatic capacity (defined by its genetics) (Doelle, 1975; Alberty, 2003). On the other hand, microbial diversity of the respiratory chain is determined by the redox potential values of the electron donor and acceptor couples, the genetic information of the organism, the ability of the organism to regulate the synthesis of redox components according to the needs of the cell under different growth conditions, and the extent to which the respiratory chain has to interact simultaneously or sequentially with more than one donor and/or acceptor (Anthony, 1988). Taken together, this leads to the question: how diverse would the 'redox profile' be across microorganisms? This conceptualisation of cellular metabolism is similar to conceptualising biogeochemical processes in the environment (Falkowski et al., 2008), but focusing on the microorganisms as the drivers of the processes, a view shared with Bar-Even et al. (2012) and Schoepp-Cothenet et al. (2013). Assuming that the microorganisms in any given environment have adapted to their own environmental limitations, the diversity of metabolic capacities in light of this thermodynamic constraint could then be compared. What are the implications of having a wider or narrower redox profile? Would that be reflected in their physiology and/or metabolism? Furthermore, how does this affect metabolic interactions between microorganisms?

### 1.3 Metabolic-based interactions: syntrophy

Metabolic interactions are ubiquitous in microbial communities (Ponomarova and Patil, 2015) and both play an important role in healthcare (Goodman and Gardner, 2018; Hendrickx et al., 2006; Hendrickx and Mergeay, 2007; Lee and Hase, 2014; Lof et al., 2017; Rodriguez-Concepcion et al., 2018; Sung et al., 2017; Yang et al., 2016b) and in driving important biogeochemical processes (Morris et al., 2013; Falkowski et al., 2008; Sañudo-Wilhelmy et al., 2014), such as the sulphur cycle and nitrogen fixation.

There are multiple types of metabolic interactions (Figure 1.5, modified from Großkopf and Soyer (2014)). Commensalism refers to an organism benefiting while the other is unaffected, while amensalism is the opposite, where one organism is harmed while the other is unaffected (Hartel, 2005), such as in food chains (Freilich et al., 2011; Rousk, 2016; Xu et al., 2011). Competition arises naturally when two organisms utilise the same substrate (Großkopf and Soyer, 2014). Unlike animal systems, predator-prey interactions in microbiology can be described as a food chain with waste inhibition; the first microorganism's waste is utilised by a second microorganism, which produces a compound that inhibits the first microorganism, normally resulting in oscillations in their growth (Balagaddé et al., 2008). Cooperation occurs with mutually beneficial interactions, of which there are different types, such as cross-feeding and syntrophy. Cross-feeding occurs when

two or more species mutually exchange metabolites, such as vitamins and amino acids (D'Souza et al., 2014; Kerner et al., 2012). Syntrophy is a special case of cooperation that takes place in energy-limited environments (Schink, 1997). It consists of a food chain where the upstream microorganism suffers thermodynamic inhibition that is alleviated by the next one, which benefits from the waste products of the first (Großkopf and Soyer, 2014). This syntrophic interaction is based on the transfer of reducing equivalents, such as hydrogen and formate, between the



**Figure 1.5** Overview of metabolic interaction motifs between two species presented from an ecological and metabolic perspective. Circles represent microorganisms, while squares represent chemical compounds (metabolites).  $\rightarrow$  indicates a stimulating or beneficial interaction, while  $\dashv$  indicates an inhibitory interaction. Figure modified from Großkopf and Soyer (2014).

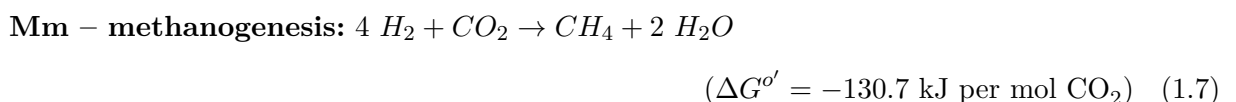
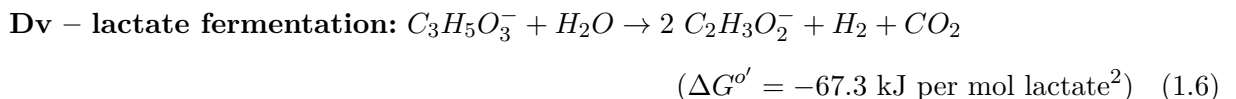
microorganisms and is also termed “interspecies electron transfer (IET)” (Schink, 1997).

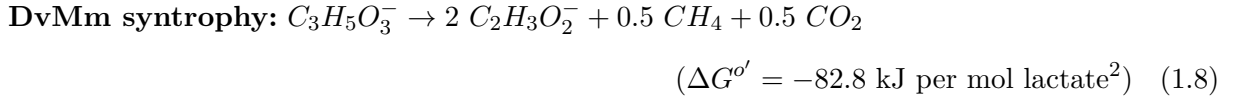
### 1.3.1 Syntrophic interactions and interspecies hydrogen transfer (IHT)

Syntrophic interactions have, therefore, a strong link with thermodynamics (Großkopf and Soyer, 2014; Schink, 1997; Stams and Plugge, 2009). Syntrophy is a key process in methanogenic and sulphate-reducing environments, as it enables microorganisms to take advantage of the metabolic capabilities of their syntrophic partner to overcome energy barriers and oxidise compounds they would otherwise not be able to (Stams and Plugge, 2009). Particularly, syntrophy has been observed through interspecies hydrogen transfer (IHT), in which  $H_2$  serves as the electron carrier between the microorganisms. The syntrophic metabolism via IHT functions well as long as the  $H_2$ -utilizing microorganisms maintain a low concentration of  $H_2$  so that it does not impose thermodynamic inhibition on the  $H_2$  producer (Zhao et al., 2017b). Hence, syntrophy, and especially IHT, is a common interaction in methanogenic communities (Bryant et al., 1967; Hamilton et al., 2015; Kouzuma et al., 2015a; Phelps et al., 1985; Schink, 1997; Stams and Plugge, 2009).

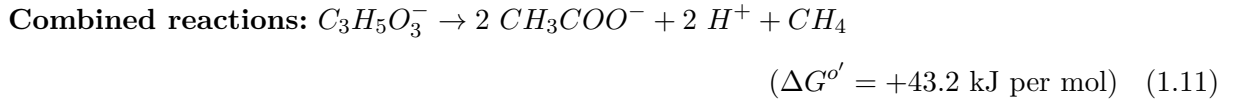
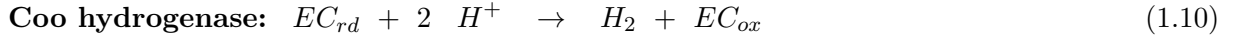
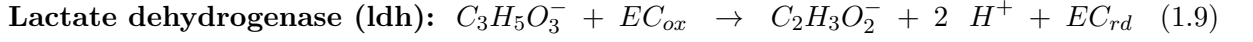
### 1.3.2 Syntrophy case study: DvMm coculture

Here a known syntrophic coculture between *Desulfovibrio vulgaris* (Dv) and *Methanococcus maripaludis* (Mm), which is based on IHT (Walker et al., 2009) and is referred to as “DvMm” throughout this work, is presented. Dv is a representative Gram (−) sulphate-reducing bacterium that couples the oxidation of lactate ( $C_3H_5O_3^-$ ; or ethanol) with the reduction of sulphate ( $SO_4^{2-}$ ) to sulphide ( $S^{2-}$ ). However, in the absence of  $SO_4^{2-}$ , Dv, as all sulphate-reducing microorganisms in general, oxidise lactate (or ethanol) and produce acetate ( $C_2H_3O_2^-$ ),  $CO_2$  and  $H_2$ , which is accumulated (Equation 1.6). However, the accumulation of  $H_2$  thermodynamically inhibits Dv’s growth. Therefore, sustained growth under these conditions is only possible through the syntrophic association of Dv with  $H_2$ -consuming partners (Walker et al., 2009). A possible partner is Mm, a hydrogenotrophic methanogen archaeon. As the name suggests, Mm uses  $H_2$  to reduce  $CO_2$  to methane ( $CH_4$ ) (Equation 1.7). Hence, Dv provides Mm with the  $H_2$  required for methanogenesis, while Mm alleviates Dv from thermodynamic  $H_2$  inhibition.





The lower free energy available for the syntrophic growth (Equation 1.8) is determined primarily by Dv's energy cost of the two-electron oxidation of lactate to pyruvate and  $H_2$  (Equation 1.11):



where  $EC_{ox}$  and  $EC_{rd}$  refer to an unknown oxidised and reduced electron carrier, respectively, as reported by Walker et al. (2009).

Further studies showed that this syntrophy is only possible through a genetic alteration in the membrane-bound Coo hydrogenase, which enables Dv to produce sufficient intermediate  $H_2$  for sustained Mm growth and, hence, syntrophy (Großkopf et al., 2016).

### 1.3.3 Syntrophy and direct interspecies electron transfer (DIET)

Direct interspecies electron transfer (DIET) has been found in syntrophic cocultures between *Geobacter* species (Summers et al., 2010) or between *G. metallireducens* with *Methanosaeata* (Rotaru et al., 2014b) or *Methanosarcina* (Rotaru et al., 2014a) species. The electron transfer was achieved by electrically conductive pili (also called nanowires), which have been extensively reviewed (e.g. Barua and Dhar, 2017; Kouzuma et al., 2015b,a; Kumar et al., 2015; Lovley, 2017b; Park et al., 2018; Shi et al., 2016) and surface cytochromes (Chong et al., 2018; Lovley, 2017b; Xu et al., 2016, e.g.). Conductive materials, such as granular activated carbon (GAC), biochar, carbon cloth and magnetite, can promote DIET in defined cocultures (Zhao et al., 2017b). Furthermore, DIET can occur by the attachment of syntrophic partners to conductive materials for interspecies electron exchange, without the production of pili or cytochromes, and, therefore, the conductive materials can promote DIET between syntrophic partners, providing an alternative to IHT (Kato et al., 2012; Zhao et al., 2017b). DIET has been used to enhance and stabilise anaerobic digestion (Lovley, 2017a), the process of breaking down organic waste, such as food waste and cow slurry, to produce methane. It is believed that electric interactions between conductive minerals and

---

<sup>2</sup> Calculated using the concentrations observed during steady state (4 mM lactate, 26 mM acetate,  $2.5 \times 10^{-5}$  atm  $H_2$ , 0.05 atm  $CO_2$ , 0.0006 atm  $CH_4$ ; temperature, 310 K) (Walker et al., 2009)



microbes may contribute greatly to the coupling of biogeochemical processes (Kato et al., 2012).

## 1.4 Electric interactions between conductive materials and microbes

As mentioned before, electron flows are inherent to the microbial metabolism in the form of redox reactions (Rabaey, 2010). Therefore, it cannot come as a surprise that extracellular electron transfer (EET) is not exclusive to syntrophic microorganisms. A range of electroactive microorganisms, sometimes referred to as electrochemically active bacteria (EAB Chang et al., 2006), have been identified which can use insoluble electron acceptors outside the cell to drive the respiratory chain through EET (Rabaey, 2010; Yong et al., 2017). An example of an insoluble electron acceptor would be the use of an electrode in an electrochemical system, which was first reported by Potter (1911).

### 1.4.1 Electroactive microorganisms and extracellular electron transfer (EET)

When studying electroactive microorganisms, electrodes are usually used as insoluble electron donors or acceptors. These organisms exhibit a range of EET mechanisms between the cell and the electrode. These include direct electron transfer by cellular components in the outer membrane (e.g. *c*-type cytochromes), direct long-range electron transfer with the use of conductive pili or nanowires, as previously mentioned, and indirect electron transfer facilitated by the use of soluble electron carriers or shuttles, both organic (e.g. flavins and cytochromes) or inorganic ( $H_2$  and iron) (Habermann and Pommer, 1991; Kumar et al., 2017; Yong et al., 2017). The flow of electrons can be achieved in both directions, i.e. from the microorganism to the insoluble electron acceptor or from an insoluble electron donor to the microorganism (Tremblay and Zhang, 2015). These types of microorganisms are also called anode- and cathode-respiring microorganisms Chong et al. (2018); Croese et al. (2011); Gimkiewicz and Harnisch (2013); Rabaey et al. (2010); Rittmann (2017).

The microorganisms most commonly used in applications using electric interactions include, but are not limited to, members of the genera *Pseudomonas*, *Shewanella*, and *Geobacter* (Yong et al., 2017). *Shewanella oneidensis* MR-1 has served as a model system to elucidate the Mtr pathway, an electron transfer conduit consisting of cytochromes and structural proteins that catalyses the electron flow from cytoplasmic oxidative reactions to electrodes (Ross et al., 2011). *Geobacter* species, such as *Geobacter sulfurreducens* and *Geobacter metallireducens*, have been used as model systems for conductive pili (Liu et al., 2005; Kouzuma et al., 2010; Park et al., 2018; Rabaey et al., 2005; Sund et al., 2007).

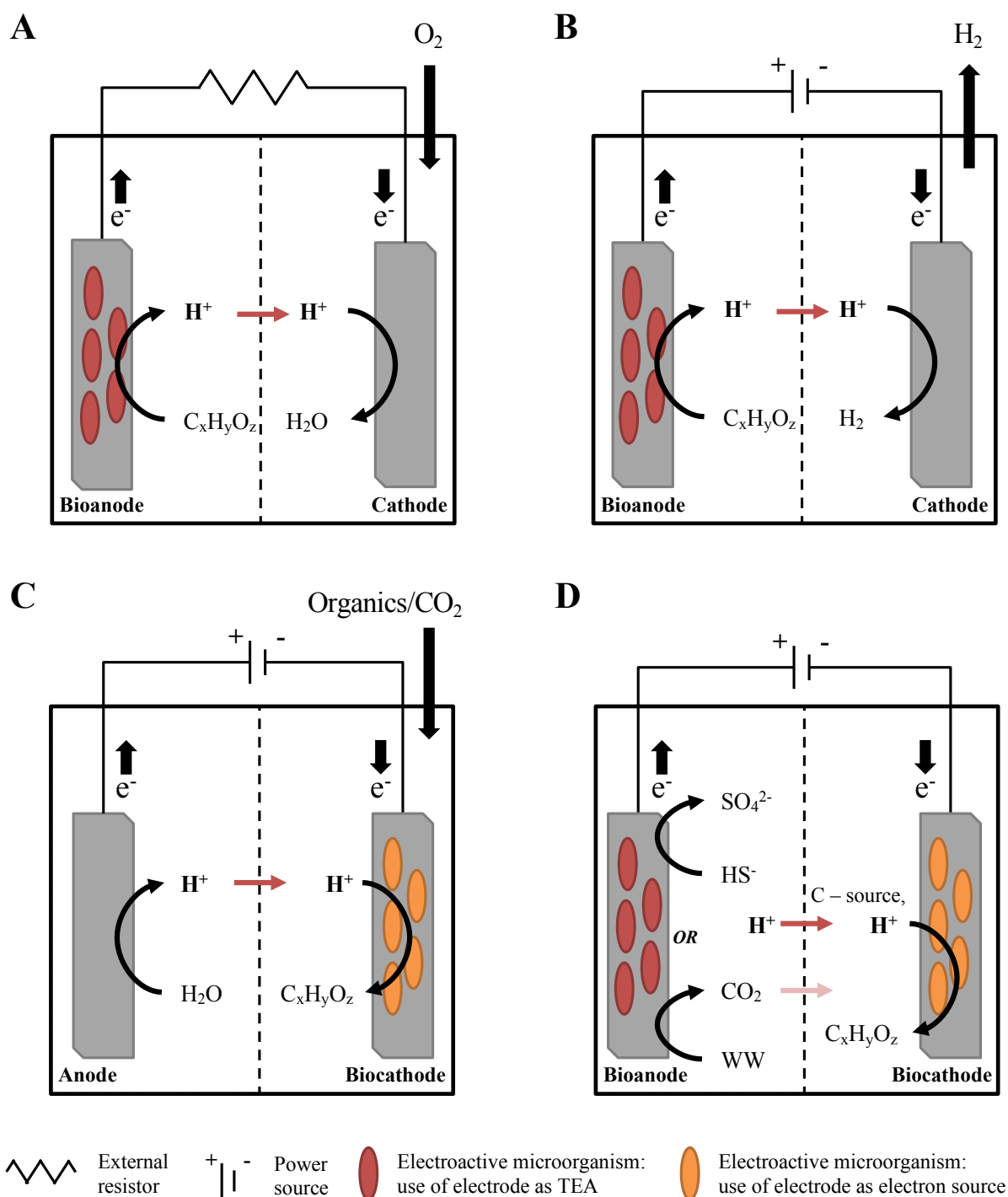
## 1.5 Bioelectrochemical systems (BES)

Electroactive microorganisms have been used for a range of applications called bioelectrochemical systems (BESs), which have been extensively reviewed (e.g. Babauta et al., 2012; Ghangrekar and Chatterjee, 2017; Rittmann, 2017; Shemfe et al., 2018; Yu, 2015). In BESs, the microbes are cultivated on electrodes, which can be thought of as an insoluble electron acceptor (microorganisms use the electrode as an electron sink; called bioanode) or donor (microorganisms use the electrode as an electron source; called biocathode). Depending on the configuration and method of application, BESs can be described as microbial fuel cells (MFCs), microbial electrolysis cells (MECs), and microbial electrosynthesis cells (MESCs) (Kumar et al., 2017). An overview of these BESs is presented in Figure 1.6.

MFCs use microorganisms to generate power by coupling their substrate oxidation ( $C_xH_yO_z$ ) with  $O_2$  reduction to  $H_2O$  at the cathode (Figure 1.6A). The electrodes are connected through an external resistor and the voltage drop across the resistor is usually measured (refer to the following reviews: Chen et al., 2017; González Del Campo et al., 2016; Logan, 2009; Ieropoulos et al., 2016; Santoro et al., 2017; Wei et al., 2011). MECs have been used to produce biohydrogen at the cathode (Figure 1.6B). Microorganisms colonise the anode (bioanode) and transform chemical energy into electrical energy coupling the oxidation of organic substrates and providing the electrodes for  $H_2$  production (normally through a potentiostat (power source)) (Kumar et al., 2017). Extensive reviews have been written on the subject (e.g. Yu et al., 2018; Lu and Ren, 2016; Kadier et al., 2016a,b; Cotterill et al., 2015). MFCs and MECs systems have been coupled waste water (WW) treatment as the microorganisms degrade the organic content and hence reduce the biochemical oxygen demand (Rittmann, 2017).

MESC aim to produce biocommodities by a range of different methods (Figure 1.6C). The most common one is to couple it with renewable power sources (wind turbine, solar cell, etc.) and divert their overproduction of power to microorganisms that can utilise it to produce value-added chemicals (Tremblay and Zhang, 2015). Therefore, MESC usually have a biocathode electrode and are connected to a power source, such as a potentiostat (as in Figure 1.6C). Alternatively, MESC systems can be coupled with other BES strategies to maximise their application and substitute the use of a potentiostat as a power source with a bioanode. An example is presented in Figure 1.6D as a MEC–MESC system (Tremblay and Zhang, 2015). MESC reviews include (Buonomenna, 2016) and (Sadhukhan et al., 2016). Shin et al. (2017) reviewed genetic modifications that have been used in MESC for biocommodities.

Each BES type can be found in different configurations, including single- and dual-chamber reactors, but they share the same operating principles (Kadier et al., 2016a). Thus BESs can be considered a feasible approach for bioremediation while simultaneously producing electricity and



**Figure 1.6** Schematic of different bioelectrochemical systems (BES) where different processes are driven by electroactive microorganisms. The BES differ on their electronic configuration and use of (bio)electrodes (grey bar). Here they are depicted as consisting of an electrochemical cell made up of two compartments (half-cells) separated by a membrane (not all BES have two compartments; see the main text for a clarification). **A** Microbial fuel cells (MFCs) are used to generate electrical power coupled with oxygen reduction at the cathode. MFCs are the only system where  $O_2$  is used. **B** Microbial electrolysis cells (MECs) are used to produce biohydrogen by reducing protons as a fuel alternative. **C** Microbial electrosynthesis cells (MESC) utilise microorganisms for the production of biocommodities by the reduction of  $CO_2$  or organic small compounds. **D** MESC coupled with a MEC (MFC-MESC) for the oxidation of sulphide or waste water (WW) treatment (on MFCs half-cell) and simultaneous production of organic compounds. WW, waste water;  $C_xH_yO_z$ , generic form for organic compounds. Figure adapted from Han et al. (2013); Rittmann (2017); Tremblay and Zhang (2015).

other value-added chemical products (Kumar et al., 2017). Additionally, BESs could provide a tool to help further our understanding of EET and to provide insights into the relationship between organisms EET, their metabolism and the ‘redox profile’.

## **1.6 Challenges remaining with electronic interfacing of microorganisms**

Although multiple BES strategies have been developed and model systems have been established for the Mtr pathway and conductive pili, still the EET mechanisms remains largely unknown. For instance, the archaeal electrical connections facilitating methane production and consumption based on DIET and the mechanism(s) for both long-range electron transfer through nanowires and cell-to-cell electron transfer still need to be elucidated. In addition, the taxonomic and mechanistic diversity of EET in prokaryotes remains largely unexplored and the “raison d’être” of EET remains largely unknown. This leads to questions such as: what are the ecological uses for, and ramifications of, the ability to transfer electrons outside the cell or for longer distances? (Lovley, 2017a; Nealson and Rowe, 2016).

Most of the studies presented so far largely ignore the energetics of the biochemical systems underlying the BES. This leads to other questions, such as what are the energetics of these reactions? What are they determined by? Can we define them based on the knowledge of the substrate oxidation and electron acceptors? How does the electron exchange that occurs through membrane-bound electron carriers affect the energetics? If the IHT of a syntrophic coculture were to be replaced by DIET as has been proposed (Kato et al., 2012; Zhao et al., 2017b), how would the energetics be affected? Would they be the same as in the syntrophy or would the DIET mechanisms change them? Can we use syntrophy over wires to study the metabolism of organisms, e.g. measuring their respiration rates? Can we hook up different (syntrophic or other) species over wires?

## **1.7 Aims and objectives**

The focus of this PhD work was on developing an experimental tool to study syntrophic interactions electrochemically as a redox process and a computational tool to study metabolic capabilities of organisms with a view to predict their interactions, including syntrophic ones. These are interlinked as the computational tool could inform experiments carried out with the experimental one to identify species interactions based on metabolic redox reactions and thermodynamics. These can then be studied and potentially influenced through electrochemical means. While this link was not completely developed, steps were taken towards establishing the development of such tools.

The work presented in this PhD thesis has four specific aims, listed below.

1. To develop a platform to enable electrochemical experiments under strict anaerobic conditions with a large experimental design that could be used to investigate the “syntrophy over wires” hypothesis (Chapter 2)
2. To investigate the “syntrophy over wires” hypothesis incorporating electrochemistry and analytical chemistry techniques using the developed platform (Chapter 3)
3. To develop a computational tool to enable the automated, large-scale analysis of annotated genomes with metabolic information (Chapter 4)
4. To demonstrate how MetQy can be used to predict electronic microbial interactions in the form of anodic and cathodic organisms that could be used to substitute Dv and Mm when testing the “syntrophy over wires” hypothesis (Chapter 5)

### 1.7.1 “Syntrophy over wires” hypothesis

This PhD work aimed to address some of the questions outlined in this introduction. In particular, the link between IHT and DIET shown by Rotaru et al. (2014a,b) and Summers et al. (2010) was found to be interesting, raising the question “could the electron transfer between syntrophic partners, such as Dv and Mm, happen between electrodes?” Therefore, the “syntrophy over wire” hypothesis was proposed.

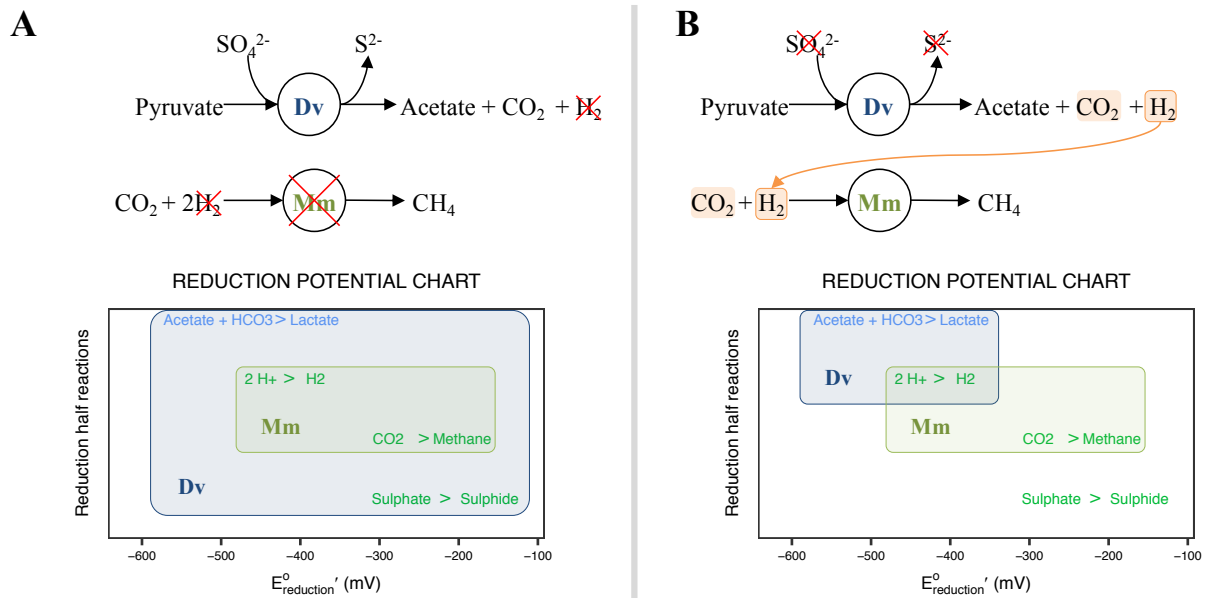
The DvMm syntrophic coculture has been well characterised both in terms of molecular biology and energetics (Section 1.3.2). Briefly, the syntrophy consists of Dv oxidising lactate into acetate,  $\text{CO}_2$  and  $\text{H}_2$  and Mm using the  $\text{H}_2$  to reduce  $\text{CO}_2$  in order to produce  $\text{CH}_4$ , thereby alleviating Dv’s  $\text{H}_2$  thermodynamic inhibition (Großkopf et al., 2016; Walker et al., 2009). Hence, the “syntrophy over wire” hypothesis refers to the interaction between Dv and Mm, whereby each microorganism would be grown on separate electrodes. Dv would use the anode as an electron acceptor and Mm would use the electrons produced by Dv to reduce  $\text{CO}_2$  into methane.

Figure 1.7 represents Dv (blue) and Mm (green) mapped onto the reduction potential chart introduced in Section 1.2. A schematic of the coculture is presented at the top, while the metabolic redox reactions relevant to the syntrophy are presented at the bottom. As mentioned in Section 1.3.2, for this syntrophic coculture to be established, sulphate must be excluded from the medium, as, otherwise, Dv’s metabolism bypasses Mm’s. This is reflected in the redox profile of the species, where hydrogen provides a metabolic overlap between species (Figure 1.7A bottom). When sulphate is removed, Dv accumulates  $\text{H}_2$ , linking both species (Figure 1.7B bottom). Furthermore, there have been reports about the capacity of electron exchange with electrodes for both Mm

(Lohner et al., 2014) and Dv (Croese et al., 2011) separately. Therefore, this argues in favour of a possible syntrophy over wires, since both species are known to be able to interact with electrodes.

### 1.7.2 Beyond a hypothesis: need for a computational tool for the discovery and design of electronic microbial interactions

Some unknowns warranting further investigation were listed in Section 1.6. Among these were the exploration of the taxonomic and mechanistic diversity of EET in prokaryotes, as well as the origin of EET (Lovley, 2017a; Nealson and Rowe, 2016). There is a long-standing research area that aims to predict functional (metabolic) traits from genomes (Green et al., 2008; Martiny et al., 2015). While many databases of genomic information exist, accessing this information and analysing it in light of specific considerations (such as redox reactions, absence/presence of electron shuttles, TEAs, and reactions that can act as thermodynamic bottlenecks) is not readily possible. A computational tool that could facilitate the integration of such information would



**Figure 1.7** Diagram of metabolic interactions between *Desulfovibrio vulgaris* (Dv, blue) and *Methanococcus maripaludis* (Mm, green) when grown in coculture. TOP – The coculture is schematically represented with circles representing the microorganisms and arrows indicating the consumption or production of chemical compounds. Red crosses refer to the absence of the chemical or the absence of growth of the microorganism. BOTTOM – The overall catabolic reactions of Dv (lactate oxidation with sulfate reduction) and Mm (H<sub>2</sub> oxidation with CO<sub>2</sub> reduction) are mapped onto a reduction potential scale referred to as reduction potential chart (Figure 1.2; Table A.1). The circles represent the extent of the organism’s metabolic range. **A** In the presence of sulphate (SO<sub>4</sub>), Dv consumes pyruvate and produces acetate and CO<sub>2</sub>, but H<sub>2</sub> is not accumulated. Mm requires H<sub>2</sub> to reduce CO<sub>2</sub> to CH<sub>4</sub> and, therefore, does not grow. The bottom representation of metabolism shows how Dv’s extends beyond Mm’s, bypassing H<sub>2</sub> accumulation. **B** In the absence of SO<sub>4</sub>, Dv produces acetate and CO<sub>2</sub> and H<sub>2</sub> is accumulated, supporting the growth of Mm. The absence of sulphate reduces the range of Dv’s metabolism to end at H<sub>2</sub> production, as can be observed in the bottom schematic, where H<sub>2</sub> serves as a link between both species. Hence, Mm has H<sub>2</sub> as a source of electrons to reduce CO<sub>2</sub>. Mm’s consumption of H<sub>2</sub> has the added advantage of relieving Dv of thermodynamic inhibition, allowing both species to grow (not represented in the Figure above).

be suitable to address these questions. This approach takes the first step in this direction and establishes a tool where genomic information can be retrieved, analysed, and visualised. It is based on existing knowledge and should, therefore, enable studies regarding the diversity of EET, as well as addressing other questions mentioned in this introduction.

# Chapter 2

## Development of an electrochemical platform

### *Electrochemistry experiments with slow growing, anaerobic microorganisms*

#### 2.1 Introduction

The experimental system described in this chapter has been designed to test the “syntrophy over wire” hypothesis presented in the Introduction. This aims to investigate whether the coculture (DvMm) between *Desulfovibrio vulgaris* (Dv) and *Methanococcus maripaludis* (Mm), which relies on molecular hydrogen ( $H_2$ ) exchange (Großkopf et al., 2016; Walker et al., 2009), can be achieved by replacing the role of  $H_2$  with electrons being exchanged over a wire. Technical challenges had to be addressed in order to test this hypothesis: an experimental platform had to be established that would support electrochemical experiments using slow-growing, strict anaerobic microorganisms in a two bioelectrode system.

The most common bioelectrochemical systems (BES), their configuration and applications were described in Chapter 1. These include the use of microorganism usually on one electrode (termed bioelectrode), which are used for specific applications, such as power generation, value-added chemical production, bioremediation, or a combination of both (refer to Section 1.5).

Multiple reactor types have been used across BES. Single chamber reactors consist of having the electrodes sealed within a bottle or cylindrical vessel with only the electrode leads protruding (e.g. Liu et al., 2005; Singh et al., 2016; Rabaey et al., 2005; Liu et al., 2004; Min and Logan, 2004; Mohanakrishna et al., 2015; Liu and Logan, 2004; Nimje et al., 2012; Santoro et al., 2013; Nishio et al., 2010; Venkata Mohan et al., 2008; Milliken and May, 2007). Some might have a



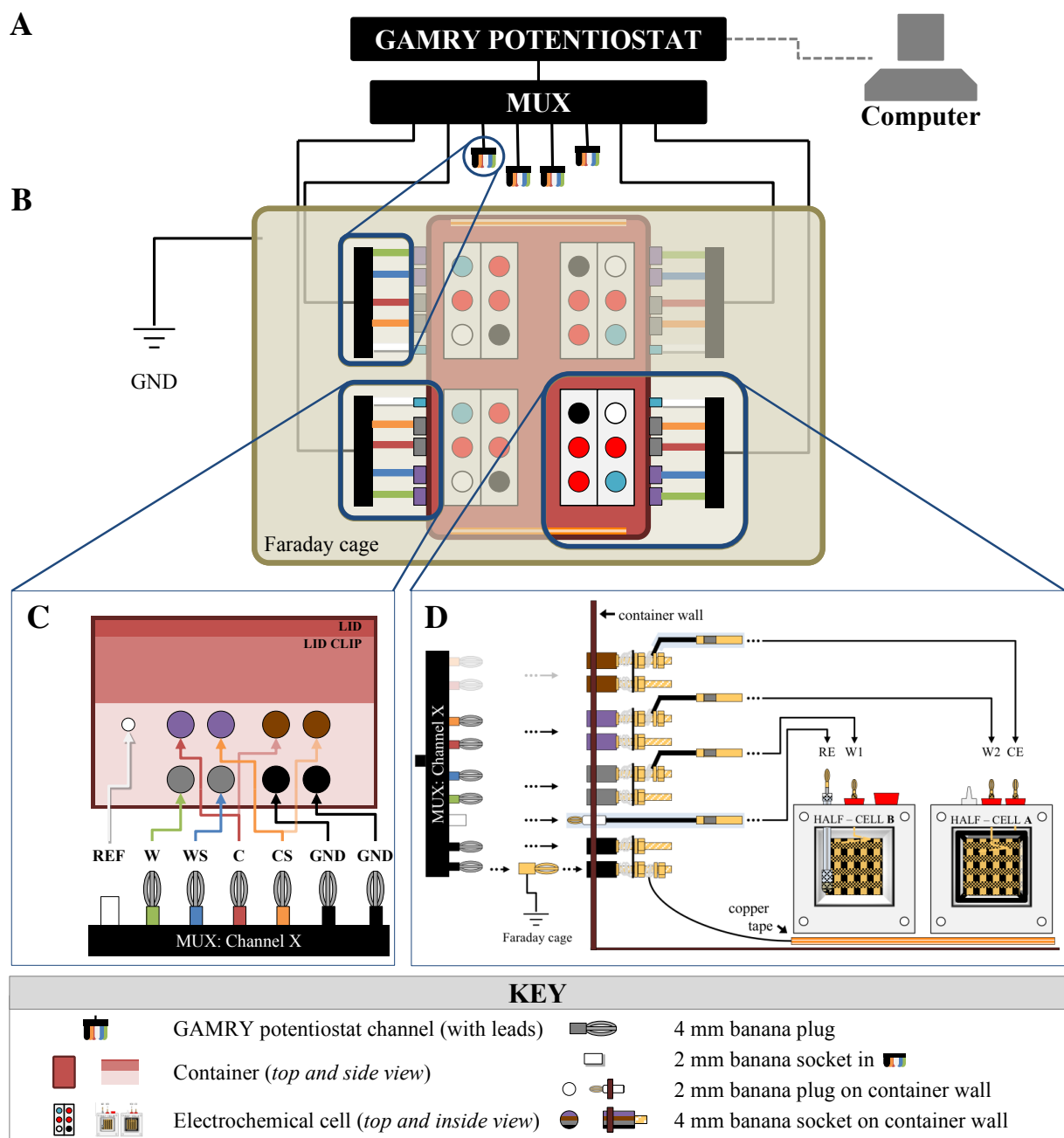
more sophisticated bioreactor design with gas spargers, stir bars, gas outlets, pH ports, etc. (e.g. Awate et al., 2017) or use alternative materials/reactors, such as terracotta caves (Gajda et al., 2015).

Two-chamber reactors (also called dual-chamber reactors) usually comprise of two bottles (H-type; e.g. Friman et al., 2012; Kim et al., 2007; McAnulty et al., 2017), glass tubes (e.g. Milliken and May, 2007) or plastic cubes (e.g. Bennetto, 1984) joined together with a membrane as separator. The designs are shared across BES types and have been widely reviewed, particularly for MFCs (Logan et al., 2006) and MECs (Kadier et al., 2016a). Other more sophisticated designs, such as flat plate MFC (Min and Logan, 2004), stacks (e.g. Trapero et al., 2017; Ieropoulos et al., 2015, 2016; Stratford et al., 2014; Hodgson et al., 2016, etc.), baffled chambered (Hu, 2008) and tubular (Hu, 2008), have also been developed.

In this chapter, the development of a platform to perform microbial electrochemical experiments to study the “syntrophy over wire” hypothesis was described (see Chapter 1, Section 1.7.1). As the two microorganisms needed to be separated, the reactor (referred to as an electrochemical cell) had to have two chambers or compartments. Hence, a cube MFC-type of reactor, similar to the one described by Kim et al. (2007) was designed (Figure 2.2). Four electrodes were required to establish a two bioelectrode system, unlike the standard three-electrode systems (Marsili et al., 2008). During the experiment, the main electrodes would be the two working electrodes (WEs). However, electrochemical characterisation of the system required a reference electrode (RE) and an auxiliary or counter electrode (CE) to be used to characterise the WEs separately. Anaerobic conditions had to be maintained for a minimum period of three weeks to ensure the microorganisms’ growth reached late exponential phase due to the organisms’ slow growth rates (see Chapter 1, Section 1.7.1). This was achieved by enclosing the electrochemical cells within containers and using chemical anaerobic atmosphere generation sachets. The containers were placed within a Faraday cage to eliminate electromagnetic noise, because low current measurements were expected.

The final experimental platform was developed in a subsystem fashion as shown in Figure 2.1. Up to four electrochemical cells were placed in a container. The container had banana connectors placed on the wall to enable electronic connections from the potentiostat to the electrochemical cells. The container(s) were, in turn, placed in a Faraday cage and connected to a potentiostat to carry out electrochemical measurements. The potentiostat was connected to a potentiostatic multiplexer (MUX) that has 8 channels to enable up to 8 electrochemical cells to be measured throughout the experiment. As our design of experiment consisted of three treatments, each with four replicates (see Chapter 3), three containers were used (four electrochemical cells were not monitored during the experiment, only characterised at the end).

The Results section (2.2) of this chapter outlines how the platform components were pre-



**Figure 2.1** Overview of the anaerobic electrochemical set-up. **A** A GAMRY potentiostat is connected to a multiplexer (MUX) that has 8 channels and is control with a desktop computer. **B** The complete anaerobic electrochemical system developed in this work. The MUX channel cables are put though a grounded Faraday cage, which holds a container (or multiple). Each container can hold up to four electrochemical cells. These are connected to the GAMRY potentiostat through connectors placed on the container wall. A channel cable is encircled at the top, with a corresponding representation of how the channel leads (left; GND lead not shown) would be connected to the connectors on the container wall (right). The chemical anaerobic atmosphere generation sachets are not shown. **C** The connection of one of the channels, “Channel X”, is shown to illustrate how each channel is connected to the container wall (viewed from the side). Each MUX channel has 7 leads that terminate in 4 mm banana plugs, except the white one (REF), which terminates in a 2 mm banana socket. The banana connectors on the wall are colour-coded according to the required GAMRY connections (Section 2.4.15.1), which are often connected together (Section 2.4.1). REF, reference; W, working; WS, working sense; C, counter; CS, counter sense; GND, ground. **D** The electronic connection between the GAMRY potentiostat and the electrodes in the electrochemical cell was achieved by having internal connections. The banana connectors on the wall are connected to 2 mm banana sockets that can be plugged onto 2 mm banana plugs protruding from the electrochemical cell. These plugs are, in turn, connected to the electrodes within the electrochemical cell.

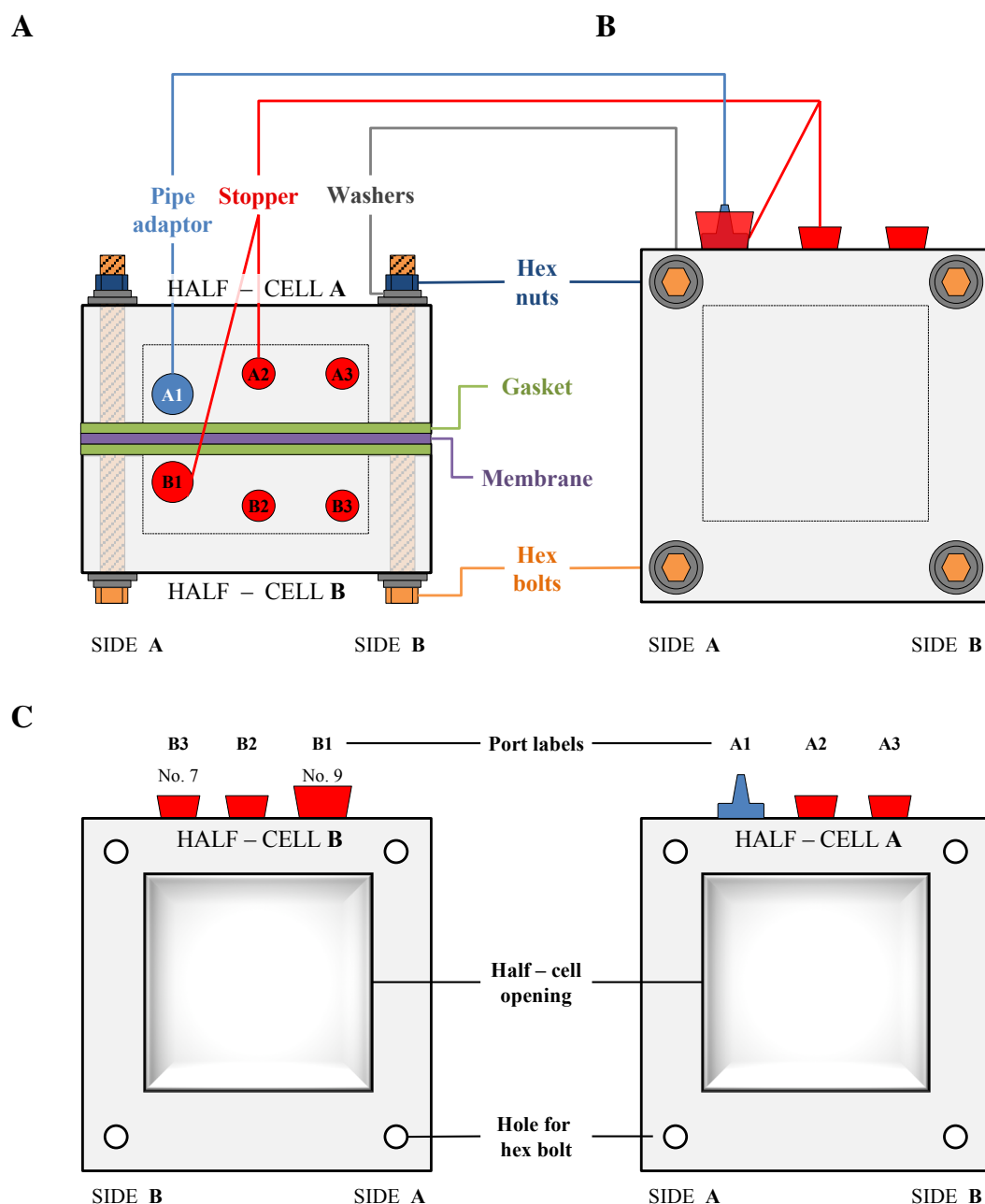
pared and assembled, while the Materials and Methods section (2.4) details the components used and steps followed for the production of all subsystem parts. Chapter 3 describes the application of this experimental platform and outlines the protocol developed to carry out the experiment required to test our hypothesis.

## 2.2 Results

The design of the electrochemical cell (Figure 2.2) was done in collaboration with Daniel Carlotta-Jones and Dr. James Stratford at the University of Warwick. The cells consisted of two half-cells and these were made of polycarbonate to allow the cells to be autoclaved. The capacity of the cells to conserve anaerobic conditions was found to be insufficiently long for the experimental requirements. Since it was not possible to determine a single origin for the oxygen contamination, it was determined that the environment surrounding the cells needed to be controlled. Multiple solutions were considered. Setting up the experiments within the anaerobic chamber was discarded due to the limited space and the need to contain the electrochemical cells within a Faraday cage. Furthermore, the use of a glove bag connected to a  $N_2/CO_2$  tank was also discarded, because the long experimental time combined with the need for wires to be connected through the glove bag were deemed impractical due to the high gas demand and the adaptation of the laboratory facilities required. Finally, it was decided to take advantage of the use of banana connectors in the potentiostat leads (see Section 2.4.1) and use a sealable plastic container, through which holes could be drilled and connectors fitted (see Section 2.4.15). Chemical anaerobic atmosphere generation sachets (Section 2.4.15.1) would be placed within the container to catalyse any oxygen that permeated inside, effectively creating an environment with limited oxygen diffusion that would increase the duration of the anaerobic electrochemical cells.

A representation of the complete system can be observed in Figure 2.1. This section describes the process required to have a working experimental platform and, by the end, the reader should have a clear understanding of the subsystems that compose it: the electrodes, the electrochemical cell, the container and the connections throughout.

Three types of electrodes were required: working, counter and reference, as mentioned before. However, two working electrodes (WEs) were used (one for each organism), making four electrodes total. The working and counter electrodes were made of carbon fibre twill coated with 25 nm and 50 nm of gold, respectively, to increase the conductivity and reduce the resistance. The reference electrode used was a silver/silver chloride (Ag/AgCl) electrode in a 4 M KCl electrolyte solution.

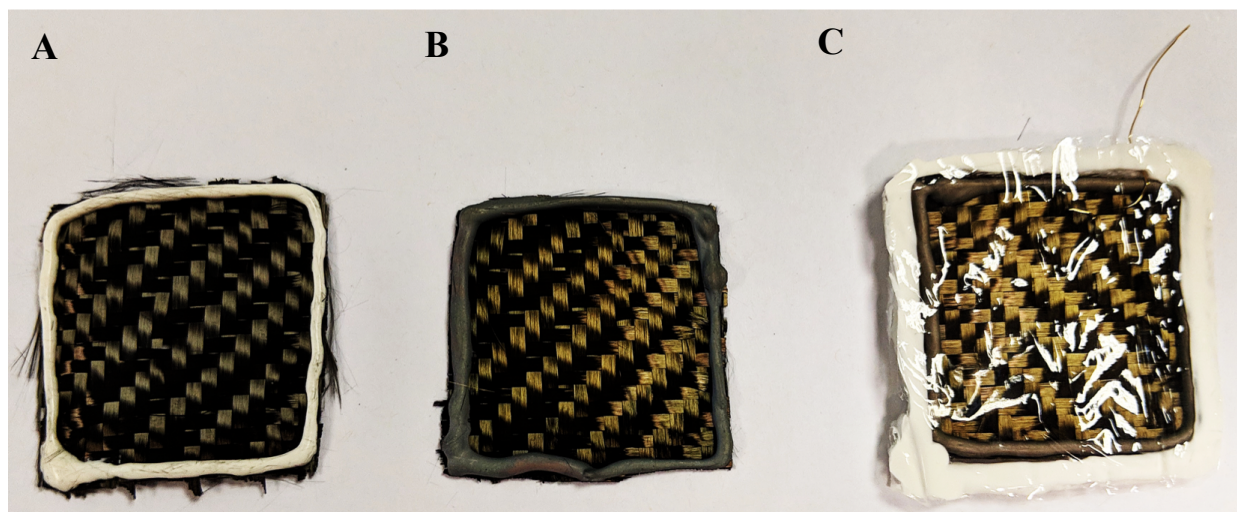


**Figure 2.2** Schematic representation of electrochemical cell. **A** Top view of the electrochemical cell. The two half-cells (A and B) were assembled with a rubber gasket (green) and membrane (purple; optional) ‘sandwich’ between. A hermetic seal was achieved by tightening the half-cells with hex bolts (orange), washers (grey) and hex nuts (dark blue). The electrochemical cell has 6 openings (referred to as ‘ports’) at the top. These are colour coded according to the component used to seal the port and contain the label used to refer to that port. There are two different types of openings: threaded, intended for pipe adaptors (blue), and non-threaded, intended for rubber stoppers (red). Note that port B1 is also threaded but was closed with a No. 9 rubbers stopper. **B** Front view of the electrochemical cell. **C** Inside view of the electrochemical half-cells. These have a square cavity at the centre, where electrodes can be placed. See Figure B.1 for the dimensions.

### 2.2.1 Working and counter electrodes

Carbon-based materials are the most commonly used for bioelectrodes due to their good biocompatibility, good chemical stability, high conductivity, and relatively low cost (Wei et al., 2011). Of the multiple types, carbon fibre twill (CFT, also known as carbon cloth) is flexible and more porous, allowing more surface area for bacterial growth (Wei et al., 2011). Thus, CFT was used as the base material. Surface coating of base electrode materials with carbon nanotubes, Pt, polyaniline, Pd, iron oxide, iron-containing graphite pastes, neutral red (NR), anthraquinone-1,6-disulfonic acid (AQDS), and 1,4-naphthoquinone (NQ) has resulted in a 0.7 – 82 –fold increase in current density (Wei et al., 2011). Coating with gold nanoparticles resulted in a 20-fold increase in current density when growing *Shewanella oneidensis* MR-1 (Fan et al., 2011). Therefore, the working and counter electrodes were both made of 4x4 cm CFT squares and both sides were coated with 25 and 50 nm gold, respectively.

CFT squares (CFTSs) were produced with the aid of a 3D printed stencil (see Section 2.4.9) and an example is shown in Figure 2.3A. The counter electrode (CE) had double the gold coating thickness of the WEs (Figure 2.3B) to simulate a larger surface area and, therefore, not limit the electrochemical reactions on the WEs. The WEs and CE were gold coated with 25 nm and 50 nm of gold sputter, respectively, with a mean thickness (95% confidence intervals) of 25.037 nm (25.018 - 25.055) and 50.061 nm (50.025 - 50.096). The CE had to be processed further to ensure its isolation from the microorganisms. This was achieved by encasing it within cellulose membrane squares, fixed together with epoxy (Figure 2.3C). A gold wire was woven into the carbon fibre before enclosing it, to allow it to be electrically connected (see Section 2.4.9.2 for the production details).

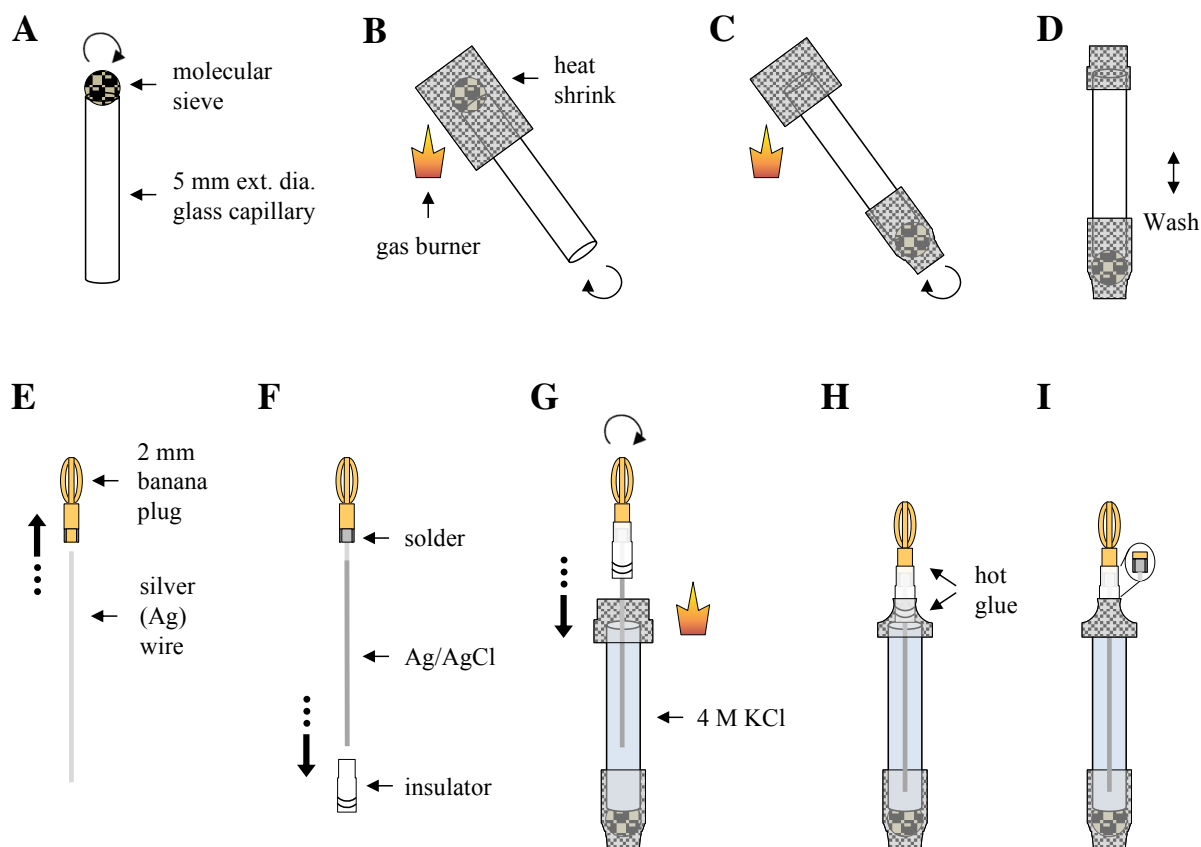


**Figure 2.3** Photographs of the CFTS-based electrodes. **A** A carbon fibre twill square (CFTS) held together with an epoxy frame (white). **B** A working electrode (WE): CFTS coated with 25 nm of gold (each side). **C** A counter electrode (CE): CFTS coated with 50 nm of gold (each side), woven with a gold wire and isolated with cellulose membrane.

## 2.2.2 Reference electrode

Commercial reference electrodes (REs) were inadequate for this application due to their size and cost, as the maximum external diameter ( $\varnothing$ ) allowed was constrained by the electrochemical cell dimensions and we failed to find a commercial RE that complied to this limitation. Therefore, a protocol to produce a silver/silver chloride (Ag/AgCl) RE was developed and is described in Section 2.4.10 and Figure 2.4. This consisted of a Ag wire coated with AgCl placed within a glass capillary filled with 4 M KCl. A molecular sieve was used to ensure contact with the electrochemical cell's medium, using inert heat shrink to fix the sieve and close the top. A 2 mm banana plug (BP) was soldered onto the Ag wire to for the internal connections (Section 2.4.15.3 and Figure 2.12).

Leakage of the electrolyte solution was observed in 23% of capillaries during the production up to the stage of Figure 2.4D. The source of leakage was mainly where the bottom heat shrink

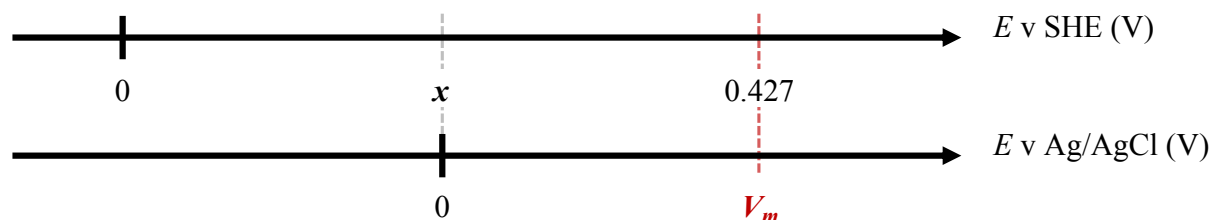


**Figure 2.4** Reference electrode production: protocol 1. **A** The molecular sieve bead was grinded on one end of a 5 mm  $\varnothing$  glass capillary tube. **B** A heat shrink was fixed holding the molecular sieve bead in place using a gas burner. The heat shrink was firmly fixed and had to be able to withstand pulling. **C** A second heat shrink was fixed at the top of the capillary without closing the opening. The heat shrink had to be firmly fixed and able to withstand pulling. **D** The capillary was washed with ultrapure water and then filled with the 4 M KCl solution. **E** A silver wire was soldered onto a 2 mm BP and then was treated in the NaOCl solution to coat the wire with silver chloride (AgCl). **F** The insulator of the 2 mm BP was put in place. **G** The Ag/AgCl wire was introduced in the capillary which contained the 4 M KCl solution. The wire was fixed in place by treating the heat shrink with the gas burner taking care not to heat the solution. **H** Hot glue was placed at the top to ensure the wire's enclosure. **I** Assembled RE. See Section 2.4.10 for additional details. BP, banana plug.

met the end of the glass capillary, due to the sharp glass cutting through the heat shrink. In future, filling of the glass capillary edges would be recommended in order to further reduce the leakage percentage. Furthermore, proper closure of the top heat shrink surrounding the insulator proved difficult as the electrolyte solution would be heated in the process, leading to evaporation of the water and, thus, change in the concentration of the KCl causing it to precipitate. Therefore, care was taken not to heat the electrolyte solution and hot glue was placed at the top, surrounding the BP insulator to ensure the closure of the opening, while minimising water loss due to evaporation. A further improvement in the manufacturing process could be implemented by introducing the insulator (see Figure 2.4F) into the capillary during the placement of the top heat shrink (see Figure 2.4C) and fixing it with heat shrink before the 4 M KCl is introduced. The insulators' top opening would permit filling of the capillary, prior to the insertion of the Ag/AgCl wire.

### 2.2.2.1 Reference electrode characterisation

As in any electrochemical system, the potential of the Ag/AgCl couple depends on the concentration of all chemical species involved, i.e. Ag,  $\text{Cl}^{-1}$  and AgCl (Lefrou et al., 2012). Therefore, it was necessary to determine the potential of each RE manufactured against the standard hydrogen electrode (SHE) due to production variations. This was achieved by using a redox buffer with known chemical species concentrations and thus known potential (427 mV v SHE). See Section 2.4.11 and Figure 2.18 for the method details. Figure 2.5 shows how the two potential reference scales are mapped. The voltage between the RE and a platinum wire ( $V_m$ ) was measured when both were submerged in the redox buffer. The corresponding value equivalent to 0 V v Ag/AgCl,  $x$ , (potential v SHE) was calculated by subtracting  $V_m$  from 0.427 V (see Equation 2.1 in Section 2.4.11).  $x$  is used as a translating value to facilitate the conversion of potential values across references. This value allows potentials using the RE to be set relative to the SHE scale, making measurements comparable. The mean (standard deviation; sd) voltage measured was 0.255 V (0.00248). The



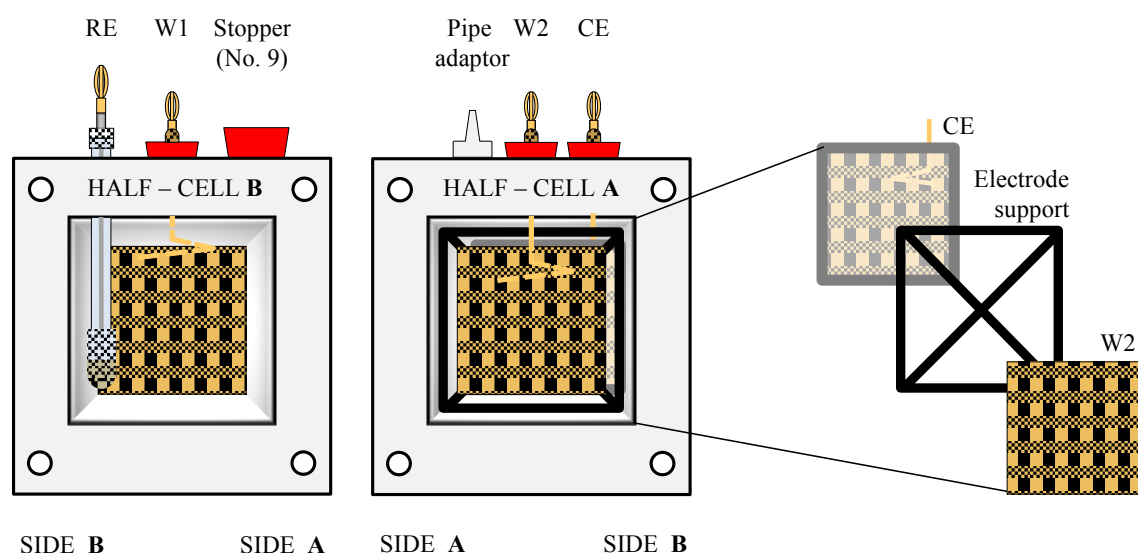
**Figure 2.5** Reduction potential translation across reference systems: Ag/AgCl and standard hydrogen electrode (SHE). The reduction potential value measured between the Ag/AgCl RE and the platinum wire in the redox buffer ( $V_m$ ) was used to determine the reduction potential value of the RE ( $x$  v SHE) that translates to 0 V v Ag/AgCl.  $x$  facilitates the translation of the potential values of the RE required to have set potential values v SHE. This is of particular importance when doing electrochemical characterisations, such as cyclic voltammetry (see Chapter 3, Section 3.4.10).

mean (sd) of the  $x$  values calculated was 0.172 V (0.00248). Once assembled and characterised, the reference electrodes were degassed prior to their introduction into the electrochemical cells to maintain anaerobic conditions.

### 2.2.3 Electrochemical cell

As mentioned before, the electrochemical cell (Figure 2.2 and 2.6) consisted of two half-cells joined with a hermetic seal obtained by placing rubber gaskets (Figure 2.15) between them and holding a non-selective membrane to separate the contents of the half-cells (components prepared as described in Sections 2.4.8 and 2.4.7). A pipe adaptor placed at the top of the cell was used for gas collection (Section 2.4.6.1) and rubber stoppers closed all other openings. The dimensions of the electrochemical cell can be seen in Figure B.1. The components details can be found in Table 2.2.

The working and counter electrodes had to be connected from the inside to the outside of the electrochemical cell. Gold was chosen over stainless steel as the wire material due to its better performance in capturing electrochemical reactions and high conductivity (see Section B.3 and Figure B.2). *Modified stoppers* (Figure 2.7 and Section 2.4.12) were produced to enable the connections, by having a gold wire protruding at the bottom soldered to a 2 mm BP at the top. The gold wire was connected to a carbon fibre electrode, while the BP was used for the internal

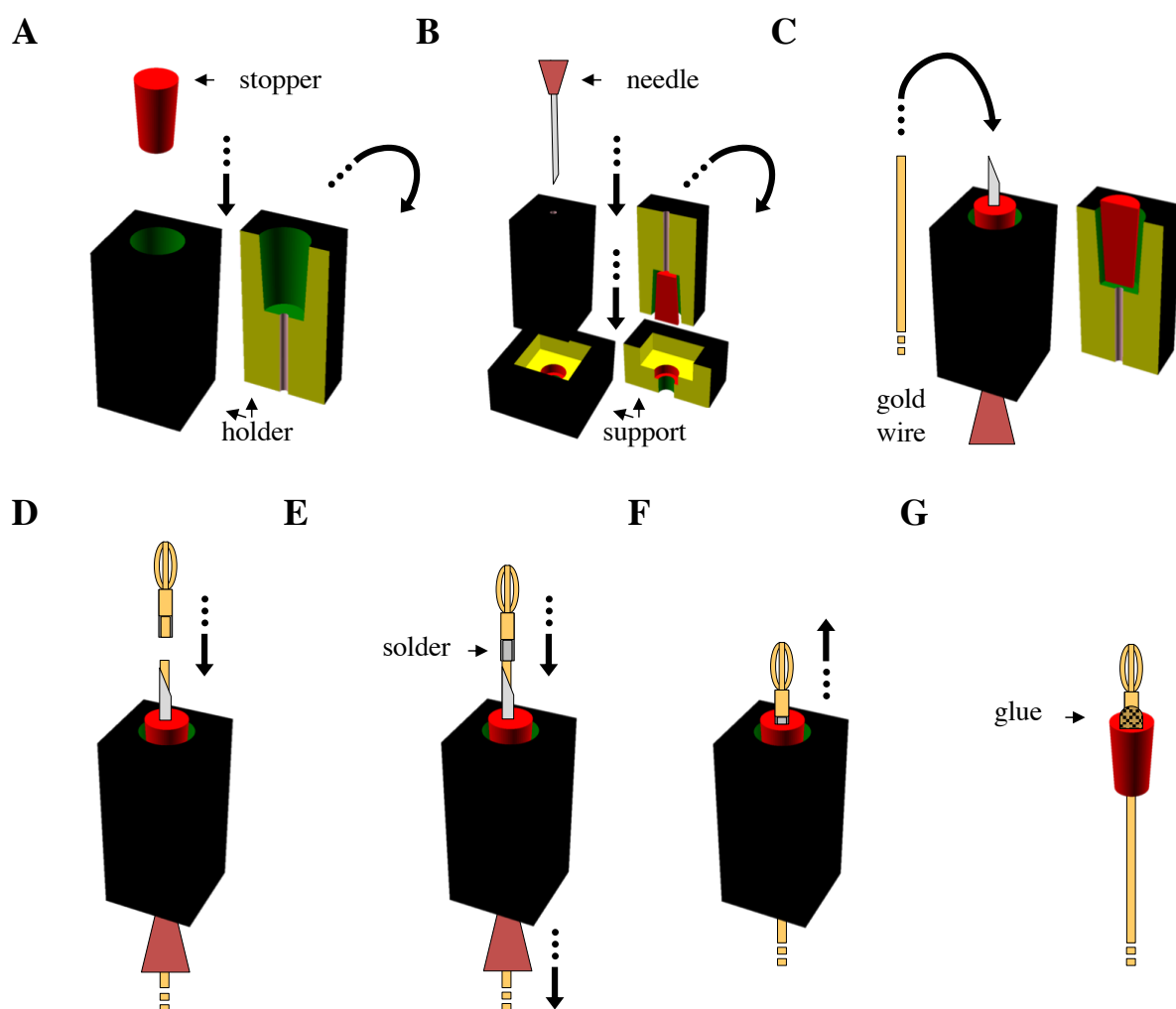


**Figure 2.6** Schematic representation of the inside of the electrochemical cell. A working electrode (W1) and the reference electrode (RE) were placed in half-cell B, while the second working electrode (W2) and the counter electrode (CE) were placed in half-cell A. W2 and CE were separated by a 3D printed “Electrode support” to ensure that they would not be in direct contact and that free diffusion of the media was achieved. Note that RE would have been placed after the assembly of the electrochemical cell, but is shown here for representation purposes. Additionally, RE was placed in the port shown or in the port occupied by the No. 9 stopper as needed.



container connections (Section 2.4.15.3 and Figure 2.12).

The positions of all the electrodes are shown in Figures 2.6 and 2.8. The W2 and CE were separated with a 3D printed “Electrode support” (black insert on the left half-cell in Figure 2.8A) to prevent direct contact and to allow free diffusion of the media (black frame with cross on Figure 2.6 half-cell A). The detailed process followed to assemble the electrochemical cells is described in Section 2.4.13 and depicted in Figure 2.19. Figure 2.8 represents the assembled electrochemical cell. Once assembled, the cells were sterilised and degassed in the anaerobic chamber (Section 2.4.14). All electronic connections underwent a conductivity test (Section 2.4.5).

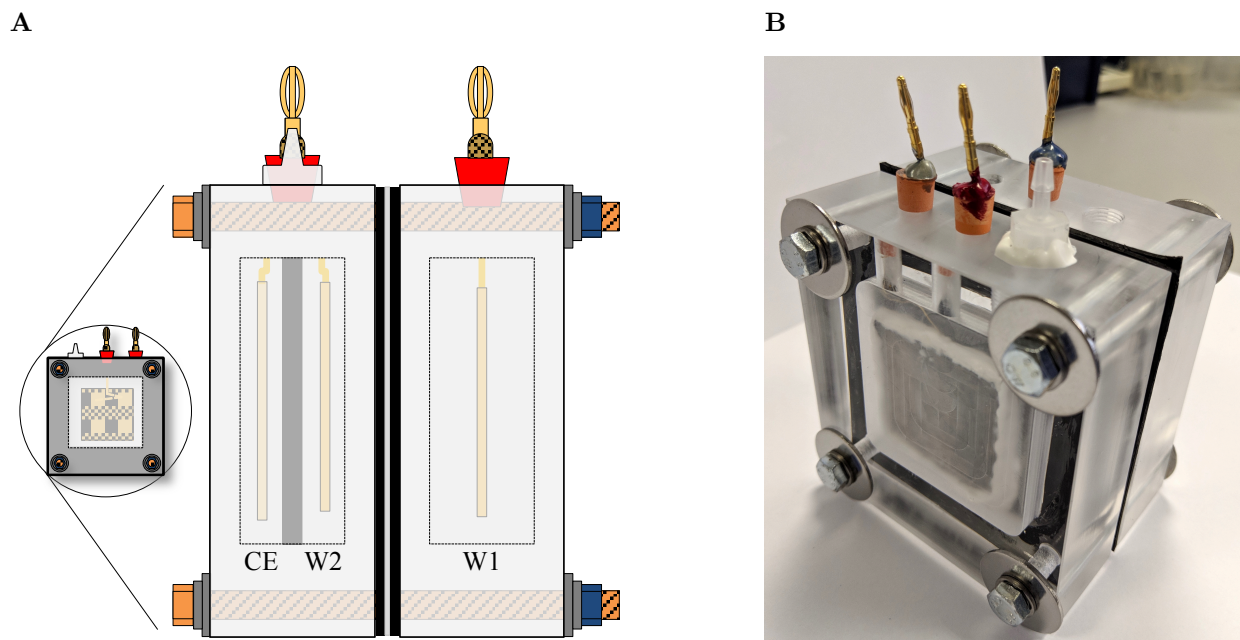


**Figure 2.7** Production of the *modified stopper*. **A** The rubber stopper was introduced into the holder (3D printed). **B** The holder was turned upside down and inserted into the support (3D printed). The needle was inserted through the holder’s opening and driven through the rubber stopper. The holder was removed from the support and turned upright. **C** The holder was removed from the support and turned upside down again. The gold wire was inserted through the needle, leaving a small protrusion on the side of the sharp edge. **D** A 2 mm BP was placed near the sharp edge of the needle and the gold wire was soldered onto it. **E** The needle was carefully removed, leaving the 2 mm BP slightly inserted into the rubber stopper to help hold it in place. **F** The rubber stopper was removed from the holder. **G** To prevent damage to the conductivity of the modified rubber stopper, structural epoxy was placed around the 2 mm BP and left to dry for 24 h. See Section 2.4.12.

### 2.2.4 Anaerobic container

The capacity of the container to maintain anaerobic conditions was tested (Section 2.4.15.1). A plugged flask filled with resazurin-containing media was placed in the container along with two chemical anaerobic atmosphere generation sachets. The medium was in direct contact with the container's environment and was visually monitored as a colour change (to pink) would indicate oxygen contamination. No colour change of the medium was observed for at least 24 days. As this met our minimum anaerobicity length requirement, mentioned in the Introduction Section of this chapter, it was concluded that the system was suitable for our experimental needs.

A connection needed to be established between the potentiostat and the electrodes within the electrochemical cells, which would be inside the container. To accomplish this, banana connectors were used. First, to achieve the connection between the potentiostat leads and the container, holes were drilled on the container wall (Figure 2.9) to place banana connectors (Figure 2.10). The Gamry potentiostat leads consist of 6 leads that terminate in 4 mm banana plugs (BPs) and the “reference” (REF) lead that terminates in a 2 mm banana socket (BS). The potentiostat connections consist of 6 4 mm BP leads are working (W), working sense (WS), counter (C), counter sense (CS), and two grounds (GNDs) as referred to by the manufacturer. To perform the electrochemical

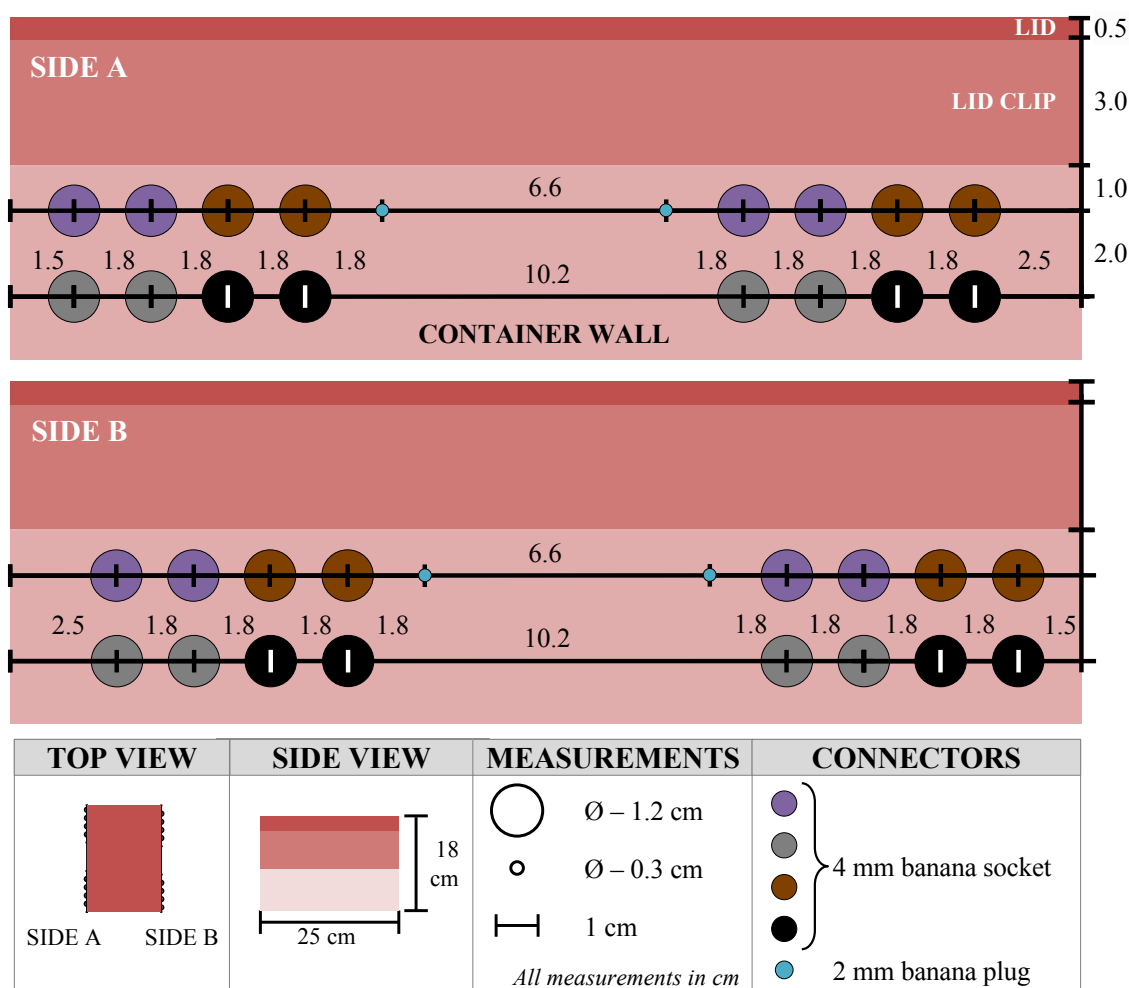


**Figure 2.8** Assembled electrochemical cell. **A** Schematic representation of the electrochemical cell assembled viewed from side A (see Figures 2.2 and 2.6). The circled areas show the view from the front, rotated clockwise on the z-axis. See Figure 2.19 for element labels. **B** Picture of an assembled electrochemical cell prior to sterilisation and degassing. The RE was introduced after the electrochemical cell was assembled and is, therefore, not shown in this figure. CE, counter electrode; RE, reference electrode; W1/W2, working electrode 1 or 2.

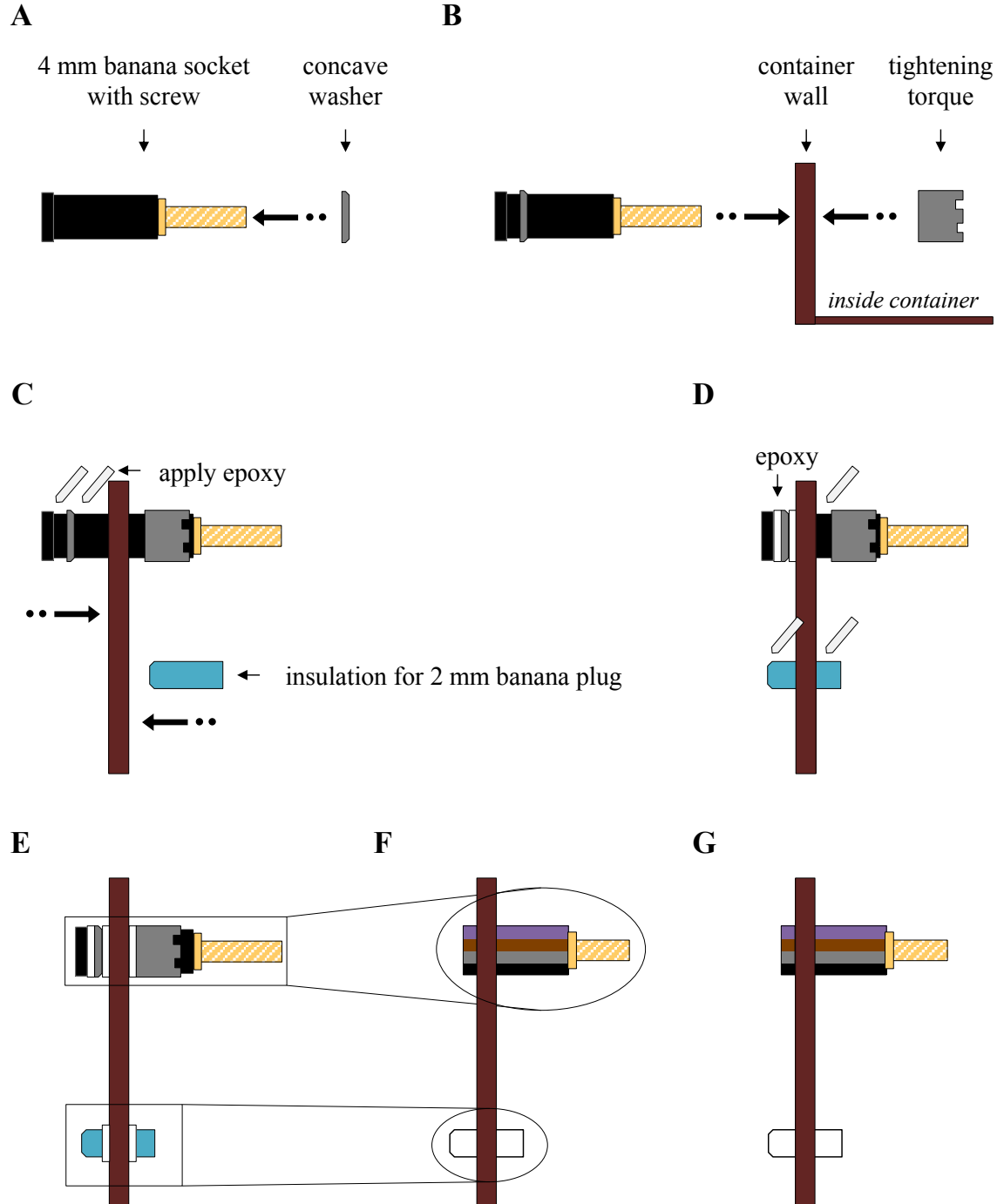
measurements required in this work (refer to Chapter 3) the 6 4 mm BP leads had to be connected in pairs as per Gamry’s connection requirements (see Table 2.1 in Section 2.4.1.1). To facilitate this, the 4 mm BP were internally shorted (Figure 2.11A).

Second, to establish connections between the container wall and the electrochemical cells, the banana connectors on the wall were connected to 2 mm banana connectors on the inside of the container (Figure 2.12). These internal connections (shaded in blue in Figure 2.11) consisted of “*wired BSs*”, 2 mm BS connected to a wire and screwed onto the 4 mm BS in the container wall, and “*wired BP-BSs*”, a *wired BS* with a 2 mm BP on the other end, inserted into the insulator placed in the container wall. The latter would be used to connect to Gamry’s REF lead. The BS of the internal connections were then clipped onto the corresponding 2 mm BS of the electrodes of the electrochemical cell, either on the *modified stoppers* or on the reference electrode. Successful connections were achieved from the container BSs to the BP on electrochemical cells, as verified with conductivity tests (Section 2.4.5).

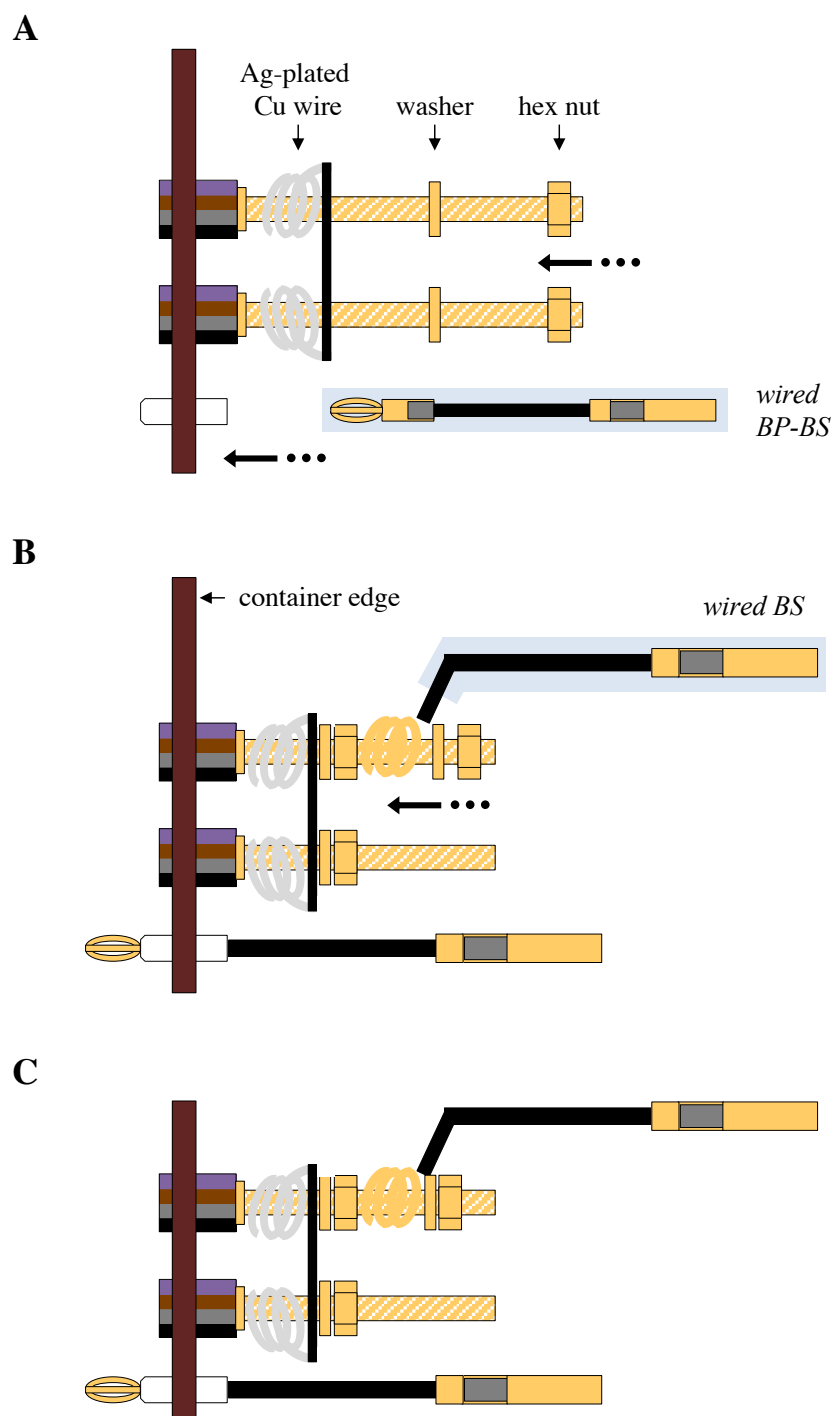
In addition to the use of a Faraday cage, a concern was the interference of electromagnetic



**Figure 2.9** Container schematic. Holes were drilled into the container’s sides to place banana connectors to enable the electrical connection from the electrochemical cells to the potentiostat.



**Figure 2.10** Container wall connectors. **A** A concave washer was placed against 4 mm banana socket lip. The black socket is shown, but the same procedure was applied to the violet, grey and brown sockets. **B** The socket was placed into the corresponding 1.2 cm  $\varnothing$  hole, facing the outside (see Figure 2.9). The torque was loosely placed at the edge of the threaded surface of the surface, in the inside of the container. **C** Epoxy was applied between the socket's lip and the washer and between the washer and container wall. This assembly was then pushed against the wall to set. **D** Epoxy was applied on the threaded area of the socket adjacent to the wall. The torque was threaded until firmly positioned. A flat screw driver was used to push the torque's indentations. **E** The insulation for the 2 mm BP was inserted into the 0.3 cm  $\varnothing$  holes. **F** The circled object will be used to represent the fixed assembled banana socket finalised, denoting the four possible socket colours. Epoxy was applied on both sides of the insulator to fix it in place. **G** Representation of the complete assembled connectors on container wall. BS, banana socket; BP, banana plug.

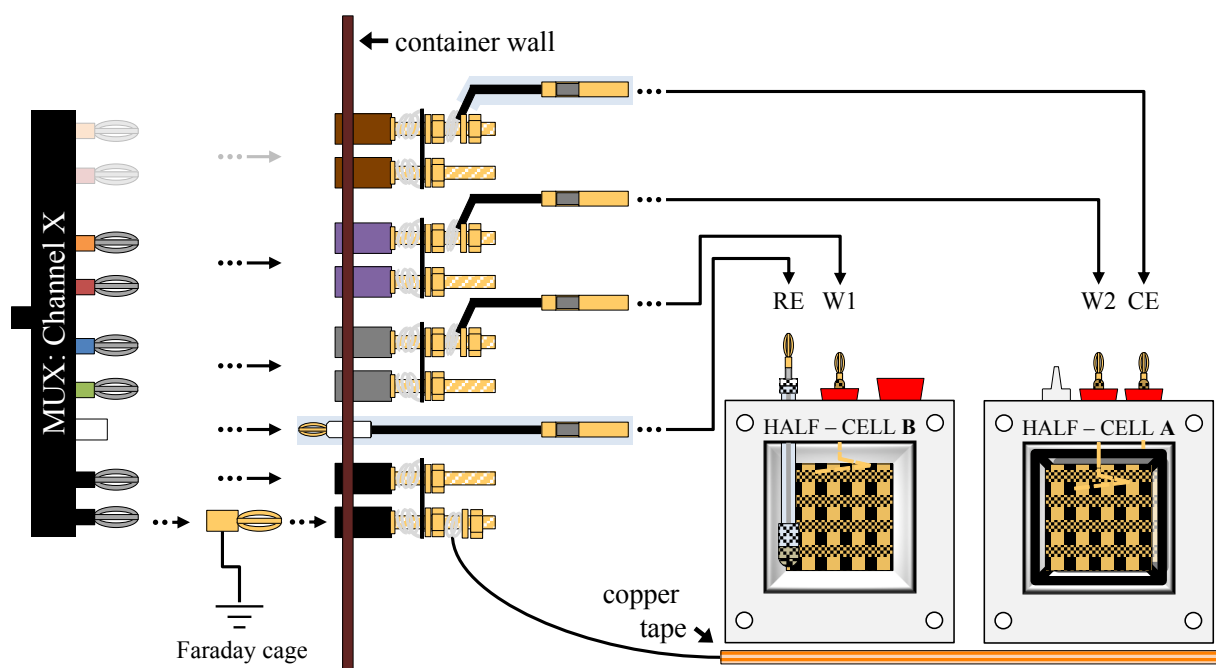


**Figure 2.11** Schematic of the electronic connections of the container. **A** Each pair of 4 mm BSs (grey, purple, brown and black, meant to be connected to W1, W2, C and ground (GND), respectively; see Figure 2.10), were connected together with Ag–Cu wire as per the connection requirements (see the main text and Section 2.4.16 and Table 2.5). A *wired BP-BS* was inserted into the BP insulator (white). **B** A *wired BS* was attached to one of the 4 mm BS pairs (all but black). **C** Complete internal connections of the container. Note that one electrochemical cell connects to four pairs of 4 mm BS and one 2 mm BP. The container has a capacity to hold up to four electrochemical cells and thus there are four sets of the connections described here per container, two on two opposite sides of the container (along the long side; see Figure 2.9). Refer to Section 2.4.15.3 and Figure 2.20 for further details about the *wired BS* and *BP-BS*. BS, banana socket; BP, banana plug.

and electrostatic noise within the container. Copper tape was placed inside the containers and grounded to avoid static charge build-up (Purcell, 2011). The copper tape was placed in both the main container and its lid, which were connected together by banana connectors, enabling discharge of electrical noises and making a pseudo Faraday cage. Figure 2.13 contains a photograph of an assembled container with its elements labelled. The container GND BSs (black) and the copper tape were successfully grounded as verified with conductivity tests (Section 2.4.5). Section 2.4.15 describes the container preparation in more detail.

### 2.2.5 System overview

After the preparation of all the subsystem components (electrochemical cell and container), the experimental platform could be set-up in the anaerobic chamber. A detailed description of the system assembly can be found in Section 2.4.17. In brief, the sterile and degassed cells were filled and inoculated according to the experimental requirements (see Chapter 3). Any opened cell ports were closed with rubber stoppers and the electrochemical cells were transferred into the container. All the electrodes were connected to the corresponding container wall connectors as summarised in Table 2.5 and Figure 2.12. Once all the connections were established, a gas collection bag was connected to each pipe adaptor for gas collection. Given that holes were drilled

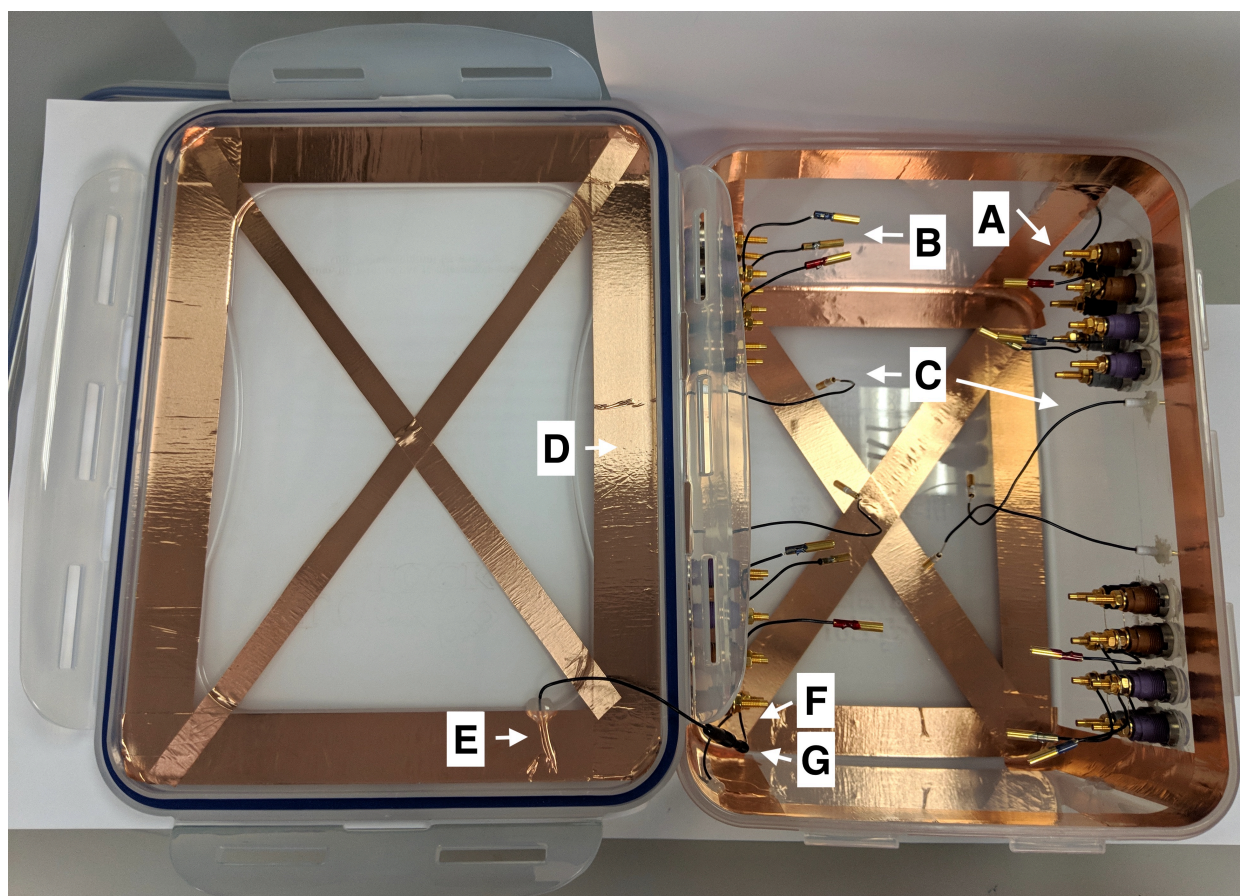


**Figure 2.12** Summary of the systems connections: from the GAMRY potentiostat to the electrodes in the electrochemical cell. **From left to right** – the leads of a representative MUX channel were inserted into the corresponding banana connectors as described in Table 2.5. An additional gold plated 4 mm BP was stacked per container to ground it by connecting it to the Faraday cage. The banana connectors have *wired BSs* and a *wired BP-BS* attached to clip onto the banana plugs protruding from the electrochemical cell, which are in turn connected to the electrodes. BS, banana socket; BP, banana plug.



to place the banana connectors in the container wall, chemical anaerobic atmosphere generation sachets (Anaerogen® pack, AN0035, Oxoid, Thermo Scientific, UK) were introduced to catalyse any oxygen that permeated into the container and thus prolong the anaerobic conditions. Once the copper tape on the lid was electrically connected to the grounded BSs in the container, the latter was closed, removed from the anaerobic chamber and placed in the Faraday cage. One black BS connector per container was connected to the grounded Faraday cage. The electrochemical cells in the containers were then connected to the potentiostat according to the experimental requirements (Section 2.4.1). The complete assembled system can be observed in Figure 2.14.

Once prepared, the containers were reused with few modifications. All internal connections were verified to ensure they still complied with the maximum resistance allowed as specified in the conductivity test (Section 2.4.5). Furthermore, after experiments were performed, the copper tape showed signs of oxidation due to the water generated by the anaerobic atmosphere generation

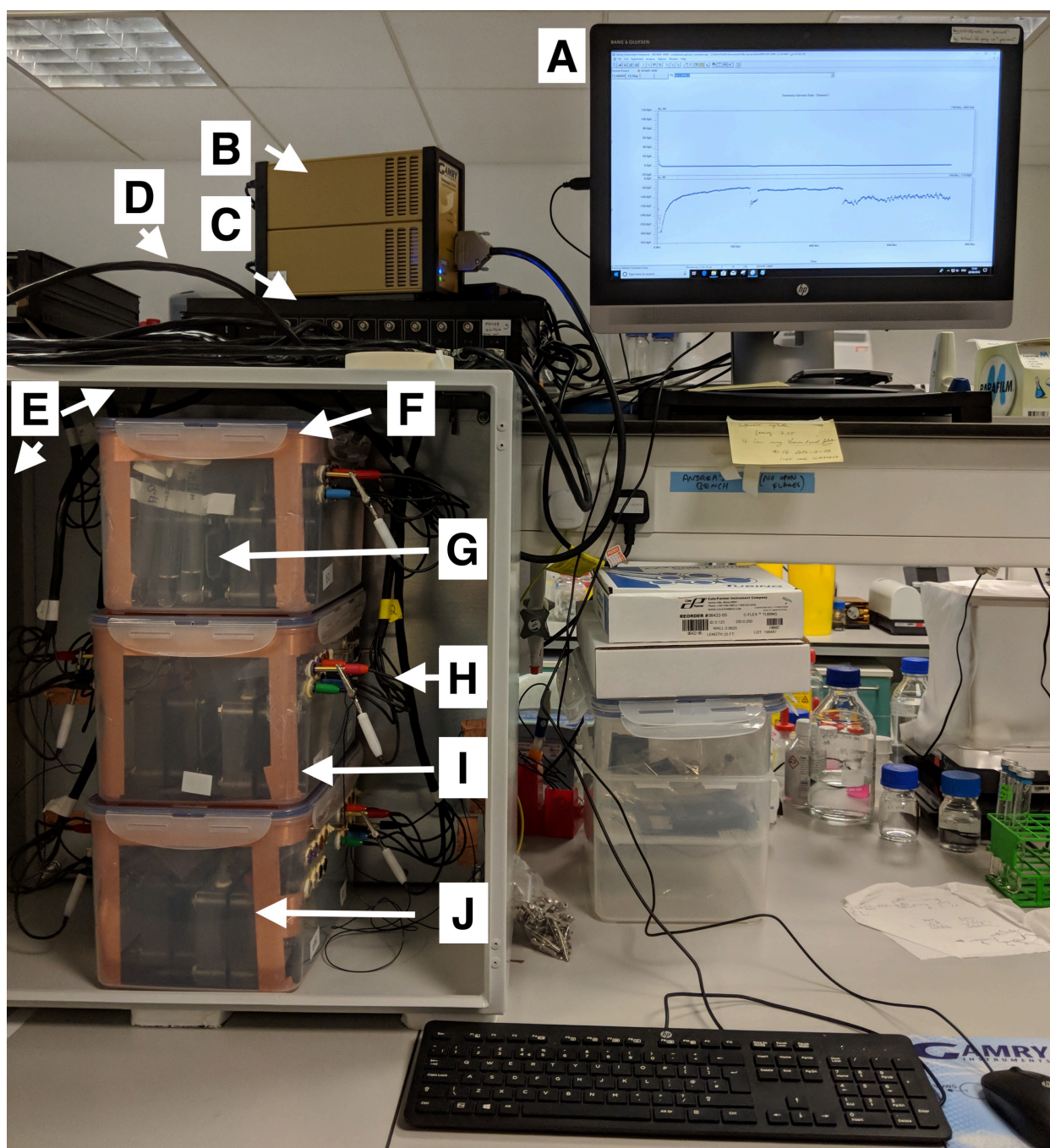


**Figure 2.13** Assembled container. **A** Wall connectors (4 mm BS). **B** *Wired BSs*<sup>1</sup> screwed onto 4 mm BS. **C** *Wired BP-BSs*<sup>2</sup> placed on insulator on container wall. **D** Copper tape. **E** *Wired BS* with insulator placed under the copper tape for lid connection, fixed with hot glue. **F** Insulated Ag-Cu wire soldered to 2 mm BP screwed on black 4 mm BS screw on container wall for lid connection. **D** Banana connectors from **E** and **F** connected to ensure grounding of the copper tape on the lid. BS, banana socket; BP, banana plug; Ag-Cu, silver plated copper.

<sup>1</sup> insulated Ag-Cu wire soldered to 2 mm BS.

<sup>2</sup> *Wired BS* soldered to 2 mm BP.





**Figure 2.14** Picture of the complete system. **A** Desktop computer with Gamry Framework software. The image displayed is the galvanic current output for channel 8. **B** Gamry Reference 600+ potentiostat. **C** Gamry multiplexer (MUX). **D** Gamry MUX channel cable. **E** Faraday cage (edge and background; lid not shown). **F** Container. **G** Test tubes within container. **H** A Gamry channel's leads connected to the connectors on a container's wall. Gamry's REF lead (white) was connected to the counter electrode (CE; set-up for EIS) by means of a crocodile clip connected to a stacked gold-plated banana plug, where Gamry's C or CS can be plugged in. **I** Copper tape inside container. **J** An electrochemical cell within container. EIS, electrochemical impedance spectroscopy.



sachets. Therefore, the copper tape was replaced as a standard procedure prior to the start of experiments.

## 2.3 Discussion

The aim of this chapter was to develop an experimental platform to enable the electrochemical analysis of anaerobic microorganisms in a two-chamber configuration. This purpose-built platform consists of a modular system, where electrochemical cells are placed within containers. Several containers can be stacked (Figure 2.14) and connected to a potentiostat with multiple channels. Most published work relies on custom made reactors, such as Bond and Lovley (2003); Milliken and May (2007); Sund et al. (2007); Friman et al. (2012); Jiang et al. (2014), and most, if not all, from Logan’s lab<sup>1</sup>, and only a handful use commercial systems<sup>2</sup> (e.g. Nichols et al., 2015; Bose et al., 2014). The former sets tend to provide very little details as to their construction (e.g. Jiang et al., 2014), hindering reproducibility and efforts to set up electrochemical experiments in laboratories without previous experience. Therefore, such a generic, open-source platform as described in this chapter can provide a system that can be adopted by those seeking to start in the field of bioelectrochemistry. The detailed preparation and assembly protocol presented in this work should also enable reproducibility across studies.

The implementation of a potentiostatic multiplexer (MUX), together with the system’s modularity, allow a higher number of electrochemical cells to be monitored and, thus, a high reproducibility within a study due to a larger design of experiment. Most common BES experiments have one replicate (e.g. Cheng and Logan, 2007; Ditzig et al., 2007; Kim et al., 2007; Wang et al., 2013, to name a few), with only a few found to have two ( $n = 2$ , e.g. Rabaey et al., 2005; Zaybak et al., 2013; McAnulty et al., 2017) or more ( $n = 3$ , e.g. Bose et al., 2014), sometimes not specifying the number of replicates (e.g. Nevin et al., 2010) or stating “replicate experiments gave similar result” (Milliken and May, 2007). The platform described in this chapter could monitor up to 8 electrochemical cells per MUX used and a given Gamry potentiostat can be connected to up to 16 MUXs<sup>3</sup>. Chapter 3 made use of one MUX to monitor two treatments with 4 replicates each. This platform has the additional advantage that the electrochemical cells are contained, making their transport and manipulation, as well as their placement in a Faraday cage, easier.

The platform has a high degree of flexibility. On one hand, the chamber area would allow different electrode materials to be used, as this is an important area within the BES research field

---

<sup>1</sup>Min and Logan (2004); Liu et al. (2004); Liu and Logan (2004); Min et al. (2005); Liu et al. (2005); Oh and Logan (2006); Logan and Regan (2006); Logan et al. (2006); Cheng and Logan (2007); Ditzig et al. (2007); Kim et al. (2007); Zuo et al. (2008); Logan (2009); Pisciotta et al. (2012); Logan and Rabaey (2012); Zaybak et al. (2013); Lohner et al. (2014); McAnulty et al. (2017) to name a few

<sup>2</sup>I am only aware of one US-based company that provides such systems:

Adams and Chittenden Scientific Glass (<http://www.adamschittenden.com/>)

<sup>3</sup><https://www.gamry.com/assets/Support-Downloads/Product-Manuals/Reference-600-Operators-Manual.pdf>

in itself (Wei et al., 2011). This platform permits the use of two bioelectrodes and thus a third electrode was introduced to act as counter electrode (CE) to both bioelectrodes (W1 and W2). The CE used here is made as the two bioelectrodes, but with a thicker gold coating instead of a larger surface area. To prevent microbes from colonising it, the CE was isolated using cellulose membrane. A test should be developed to evaluate adequate enclosure of the CE. Additionally, gas pockets formed within the CE when the electrochemical cells were filled due to a slow diffusion process, but these dispersed in time. Alternative CE materials, such as platinum wire, the most common CE material, could be tested for the purpose of this work.

On the other hand, the modular connections would allow different configurations to be implemented by changing the internal connections or connecting an external resistor across the corresponding connectors (W1 and W2) to achieve a MFC configuration (see Figure 1.6). Furthermore, crocodile clips are the most commonly used means of establishing connections; these could be clipped onto 4 mm banana plugs connected into the wall banana sockets (see Figure 2.14H), making this platform compatible with other potentiostats, volt- and multimeters. Further use of 3D printed structures could be used to define operational values, such as the distance between electrodes. Because this is an enclosed system, this platform could be used to investigate the effect of different atmospheres (e.g. different O<sub>2</sub> concentrations) on the electrochemical system under investigation.

Anaerobic condition within the containers (and, therefore, the electrochemical cells) was maintained by controlling the environment surrounding them, as the electrochemical cell's hermetic seal allowed oxygen permeation. Therefore, anaerobic conditions were achieved by placing the cells in a container, assembling the system in an anaerobic chamber and inserting chemical anaerobic atmosphere generation sachets into the container to generate a oxygen gradient between the atmosphere and the electrochemical cells. Strict anaerobic microorganism have successfully been cultured in BES before. *M. barkeri* was successfully grown for 3.5 days in a commercial two-bottle system by assembling the system in an anaerobic chamber and sparging the headspace with CO<sub>2</sub> every 24 hours (Nichols et al., 2015). *Geobacter sulfurreducens* has also been cultured in air-tight, custom-made dual-chamber glass MFC (Bond and Lovley, 2003; Reguera et al., 2006). An alternative means of maintaining anaerobic conditions has been continuous sparging with N<sub>2</sub> gas (e.g. Oh and Logan, 2006; Ditzig et al., 2007; Kane et al., 2013, the former operated for less than 10 days). In order to extend the anaerobic period, sparging with anoxic gases (e.g. N<sub>2</sub>, N<sub>2</sub>/CO<sub>2</sub>) could be implemented in the platform by introducing gas inlet and outlet ports into the container wall.

There are some main disadvantages to the platform's modularity. The first is that multiple electrical connections are required, which result in an accumulated resistance. The resistance

imposed on MFCs has been shown to affect the microbial electrochemical processes (Sund et al., 2007; González Del Campo et al., 2016) and is therefore not to be neglected. An additional step in the assembly protocol could be implemented to measure the internal resistance of the connections. This could be done by performing a “short circuit lead” test as per the application note (Gamry Instruments, 2012). Although this would not solve the issue, the resistance could at least be quantified and the use of external resistors could be used to normalise it across all connections.

The second disadvantage is that the design does not allow the electrochemical cells to be easily sampled throughout an experiment, as this required the disconnection of the container and the sampling to take place in an anaerobic environment. Moreover, implementation of chemostatic operation (or continuous mode; e.g. Rabaey et al., 2005; Zhuang et al., 2010; Marsili et al., 2008; Gajda et al., 2015; Hou et al., 2014; Yang et al., 2016a; Torres et al., 2008) would be difficult. However, pipe adaptors could be placed in the container wall, similarly to the proposal to implement gas sparging. Considerations as to the number of connections and the container structure’s stability would need to be taken into consideration.

Monitoring microbial growth would be of interest, specially for basic research like the one presented here. While the power output or current could be used to extrapolate information about the biofilm growing on the electrode(s), the researchers know nothing about microbial growth occurring in the electrolyte or medium suspension. Further work could focus on the implementation of turbidity measurements of the liquid phase, similar to the work presented in Sasidharan et al. (2018). A simple optics system relying on a LED and light–sensor pair and controlled by a microprocessor could be placed outside each half–cell and the change in light intensity monitored to quantify the medium’s turbidity.

Another physical aspect that is key for microbial growth is temperature. Out of over 52 research papers, only five (<10%) implemented temperature control. Friman et al. (2012); Nichols et al. (2015); Schmitz et al. (2015) and Kracke (2016) used water baths, while Lohner et al. (2014) used a magnetic stir–plate. However, temperature plays an important role in microbial physiology, as it influences the reaction rates (Price et al., 2001), enzymatic activities (Cohen, 2014), fluidity of the membrane and thus transport reactions (Alberts et al., 2002) and overall growth (Alberts et al., 2002). Controlling this experimental parameter would expand the range of microorganisms that could be studied in BES and its implementation should definitely be considered in future work.

Due to its high versatility, this platform can be used for a wide range of microbial electrochemical experiments and applications. This is due to the platform’s flexible configuration, which means it can be adapted to any BES configuration, specially designed for anaerobic experiments. As such, this platform is suitable to investigate our “syntrophy over wires” hypothesis. Fur-

thermore, research can be carried out to better understand extracellular electron transfer (EET) processes, the link between thermodynamics, microbial metabolism and microbial electrochemistry and electronic control of metabolism. The last two points could benefit from the four electrode system presented here (with some modifications), as a recent patent (Ieropoulos and Greenman, 2018) suggests. Ieropoulos and Greenman (2018) presented a method in which an external power source, “driver”, can help drive a MFC (working unit) by introducing a second, non-redox electrode (i.e. not a reference electrode; referred to as auxiliary electrode). The driver needs to be connected to the working and auxiliary electrodes in the same half-cell. The voltage output of the driver affects the electrochemical redox value of the half-cell’s electrolyte, which in turn affects the electrochemical performance of the system. This method has achieved an increase in power output of the MFC connected to an external power source. Furthermore, this patent provides a method of (1) controlling the redox potential of a MFC, (2) measuring the redox potential difference between half-cells (open circuit equivalent) from an operating MFC under load, without breaking the circuit and waiting for steady state, (3) exerting dynamic modulatory control of the power supplied to the MFC in response to the MFC’s performance, and (4) connecting multiple MFCs to a multiplexer, allowing the alternation of the MFCs between acting as driver or the working unit and thus enable addressing research questions involving the dynamic shift of redox conditions. This could potentially be expanded to other BES and open a whole new set of questions.

## **2.4 Materials and Methods**

### **2.4.1 Electrochemical measurements**

The equipment used to carry out electrochemical measurements was a Gamry Reference 600+ potentiostat/galvanostat/ZRA (Gamry Instruments, USA), which was controlled using Gamry Framework Software Version 7.05.5050. In order to be able to perform measurements on multiple electrochemical cells, a multiplexer (MUX; ECM8 Electrochemical Multiplexer, Gamry Instruments, USA) with 8 channels was connected to the Gamry potentiostat.

#### **2.4.1.1 Potentiostat connections**

Each Gamry MUX channel has a PC4 cable with 7 connections ending with banana connectors (see Table 2.1). These connectors were used to establish an electronic connection between the electrochemical cells and the potentiostat.

**Table 2.1** MUX channels connections

Colour <sup>1</sup>	Type	Name	Normal Connection
Blue	4 mm BP	Working Sense (WS)	Connect to working electrode
Green	4 mm BP	Working Electrode (W)	Connect to working electrode
White	2 mm BS	Reference (REF)	Connect to a reference electrode
Red	4 mm BP	Counter Electrode (C)	Connect to counter electrode
Orange	4 mm BP	Counter Sense (CS)	Connect to counter electrode
Long Black	4 mm BP	Floating Ground (GND)	Leave open or connect to a Faraday shield
Short Black	4 mm BP	Chassis (Earth) Ground (GND)	Leave open or connect to a Faraday shield

<sup>1</sup>Colour on Gamry lead; BP, banana plug; BS, banana socket. *Table modified from Gamry ECM8's manual.*

## 2.4.2 Adhesives

Two main types of adhesives were used, either structural epoxy or hot glue. Structural epoxy (Scotch-Weld DP760, 3M, UK) was used to seal joints that required low oxygen permeability or components that would be in direct contact with water (or both). The epoxy was applied using a mixer nozzle (RS Pro Epoxy Mixer Nozzle, RS Components Ltd., UK) attached to the RS Pro Sealant Gun (RS Components Ltd., UK). Once applied, it was left to dry for at least 20 h. Hot melt glue (240-12-300-CRP-TP16-RS, 12 mm transparent stick, Power Adhesives, UK) was applied with a hot melt glue gun (TEC305, Power Adhesives, UK) as an alternative adhesive.

## 2.4.3 3D printing

3D objects were designed using the OpenSCAD software v 2015.03 ([www.openscad.org](http://www.openscad.org)). The designs were exported as a stereolithography (STL) file, which was then opened through the Ultimaker Cura software version 3.3.1 to generate the g-code used by the 3D printer. The objects were 3D printed using the Ultimaker 2+ (Ultimaker B.V., UK) printer using polylactic acid (PLA, black filament, 2.85 mm, 750 g, Ultimaker B.V., UK) as the printing material with default settings.

## 2.4.4 Soldering

All soldering steps were carried out using a Weller soldering station (T0053250399N, Weller, DE) and Multicore solder (Part ID 291340, Multicore, USA) at 200 °C, unless otherwise stated. All electronic connections were tested after preparation using the conductivity test.

## 2.4.5 Conductivity test

A hand-held digital multimeter (Fluke-115, Fluke, UK) was used to verify successful electronic connections after soldering or assembly of parts by measuring the resistance across them. When 4 mm banana sockets (BS) were involved, the multimeter leads (banana plug, BP) were directly inserted into the BS. When other materials were involved (e.g. wires, banana plugs), the component was firmly pressed on the multimeter lead. Assembled components with connections with

a resistance larger than  $1\ \Omega$  were discarded when wires were involved. When the resistance was measured in circuits involving CFT, a  $50\ \Omega$  tolerance was observed.

## 2.4.6 Electrochemical cell

The electrochemical cell used is shown in Figure 2.2. This was designed to have two half-cells, to allow a membrane to be placed in-between to separate the two microorganisms. The component specifications can be found in Table 2.2.

**Table 2.2** Specification of electrochemical cell components. The ‘Part’ column contains the name used to refer to the component throughout the text. A schematic representation of the electrochemical cell can be seen in Figure 2.2.

Part	Material	Brand	Part ID
Half cell	Polycarbonate	WKH Group, UK	–
Gasket	TSEC Viton® grade A FKM rubber	The Seal Extrusion Company Ltd, UK <sup>1</sup>	–
Pipe adaptor	Polypropylene	Cole-Parmer, UK	TW-40621-28
Rubber stopper	Red rubber	Ley Holdings Ltd, UK	RB007/S
Rubber stopper	Red rubber	Ley Holdings Ltd, UK	RB009/S
Screw connectors	Polypropylene	Cole-Parmer, UK	TW-40621-28
Cellulose membrane	Regenerated cellulose	Medicell International, UK	DTV.12000.10.15
Screws	High Tensile Steel	RS Pro, UK	917-3129
Washer - big	Stainless steel	RS Pro, UK	507-2684
Washer - small	Stainless steel	RS Pro, UK	527-404
Nut	Stainless steel	RS Pro, UK	189-591
Tubing	C-FLEX®, int. dia. 0.125, ext. dia. 0.25, wall 0.0625 (in)	Cole-Parmer, USA	06422-05
Gas collection bag	Tedlar® Bags, Push Lock Valve, 1 L	Supelco, USA	24633

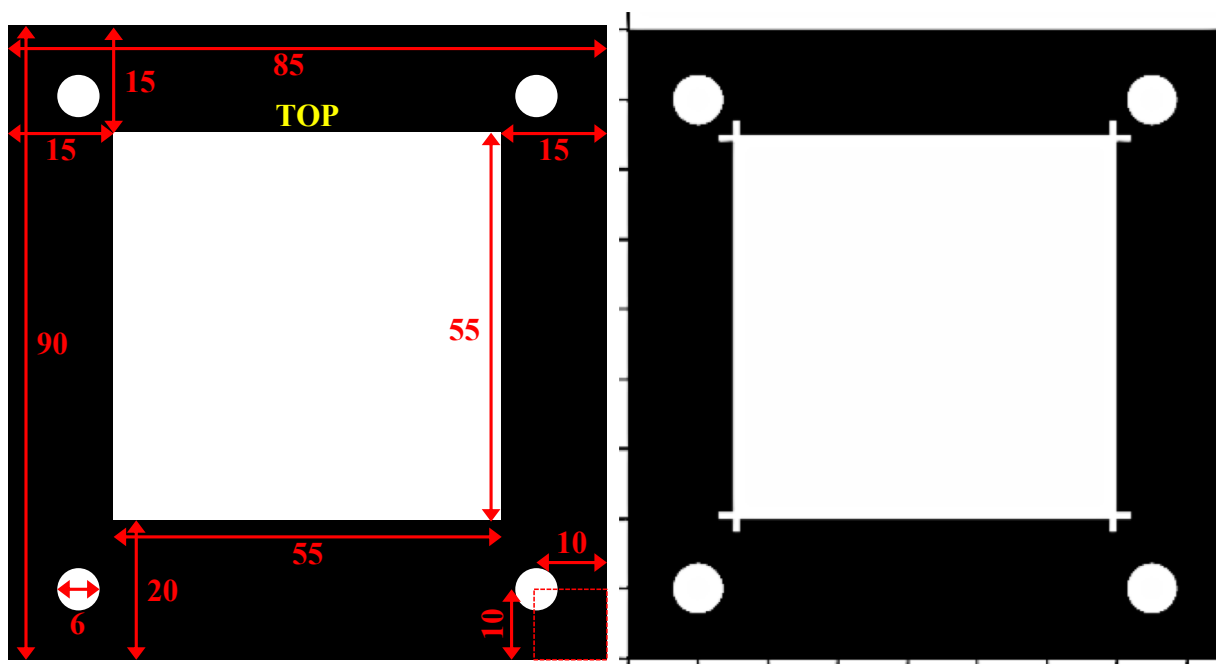
<sup>1</sup><https://www.amazon.co.uk/Viton-grade-rubber-sheet-100mm/dp/B00MPHD1BQ>

### 2.4.6.1 Addition of pipe adaptors

The pipe adaptors were screwed in place making sure that the threads were properly aligned. To decrease the oxygen permeability, the base of the pipe adaptor was sealed with epoxy.

### 2.4.7 Rubber gasket production

Rubber gaskets were required to ensure the hermetic seal of the two half-cells and the cellulose membrane. Sheets of Viton rubber were cut using a Stanley knife using a 3D printed stencil (see Figure 2.15, right and Code B.3). The holes were punctured with a 7 mm (dia.) hole punch. Like the electrochemical cell, note that the bottom frame is thicker than the top.



**Figure 2.15** Preparation of Viton rubber gasket. **Left** – measurements to cut a sheet of Viton rubber to appropriate dimensions. **Right** – Top view of 3D printed stencil produced to aid in the cutting (height = 3 mm; ticks every 10 mm). All measurements in millimetres.

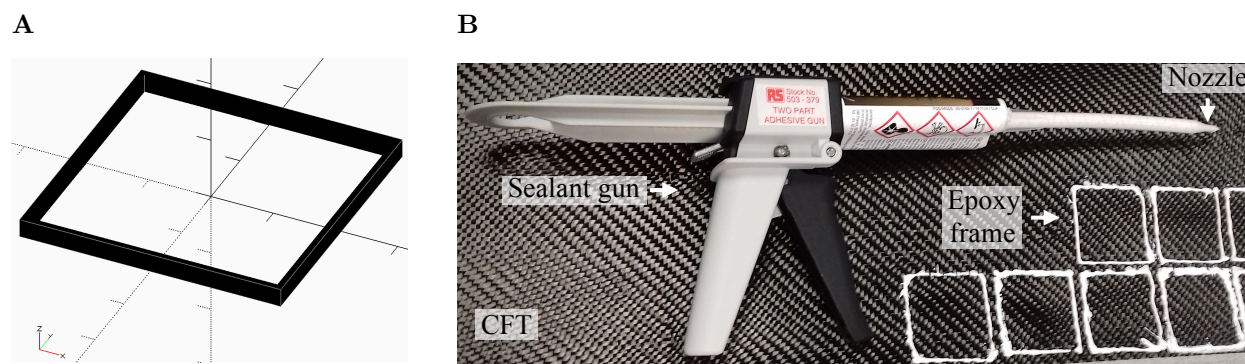
#### 2.4.8 Cellulose membrane production

The cellulose membrane was produced by cutting a 8.4 cm length of dialysis tubing (Medicell International, UK). This was then submerged into ultrapure water to hydrate it. The membrane tube was cut open using a pair of long bladed scissors. The membrane square was placed between optical tissues (Whatman N0 105 Lens Tissues, GE Healthcare, USA) to prevent fibre addition. The optical tissues were covered by paper towels and pressed flat using half-cells as weights to dry the cellulose membranes.

#### 2.4.9 Working and counter electrode manufacturing

Carbon fibre twill (CFT; Carbon Fibre 2x2 Twill (200 g/m<sup>2</sup>, High strength Carbon 3K), Polycraft, MB Fibreglass) was used as the base of the working and counter electrodes. As its structure is similar to fabric, it requires “basting” or a structure that will conserve the shape. This was achieved by making epoxy frames on one side of the CFT. Once set, the epoxy held the CFT together and prevented fibres from becoming loose. A square frame (4.0 x 4.0 x 0.3 cm, with a wall thickness of 0.2 cm), referred to as ‘electrode stencil’, was 3D printed to assist in the production (see Section 2.4.3, Figure 2.16A and Code B.2). Three electrode stencils were arranged on the CFT in an ‘L’ shape in order to allow mass production. The epoxy was pressed lightly into an inside corner of the first frame. The nozzle was then lifted while epoxy was extruded to produce a string of epoxy that could then be guided along one of the inner walls of the stencil.

Care was taken not to rub the CFT to avoid damaging it. Once the full frame had been traced on the three stencils, these were carefully lifted and placed next to a neighbouring stencil to continue the production. Any epoxy left on the stencil was immediately removed using a paper towel. The epoxy was left to dry for 24 h. Figure 2.16B shows a picture taken of two rows of epoxy frames placed on CFT, with the sealant gun used to make them. After the epoxy had set, carbon fibre scissors (Heavy duty shears for carbon fibre, William Whiteley & Sons, UK) were used to cut along the outside of the epoxy frame in a fume hood. Any loose fibres were removed. This process resulted in CFT squares ca. 4 cm<sup>2</sup> (Figure 2.3A).



**Figure 2.16** Working and counter electrode manufacturing **A** Top angled view of 3D printed stencil produced to aid in the production of the working and counter carbon fibre twill electrodes (40 x 40 x 3 mm external measurements; frame thickness = 2 mm). Ticks drawn every 10 mm. **B** Carbon fibre twill electrode base.

#### 2.4.9.1 Gold coating of electrodes

The CFT squares manufactured as described in the previous section were gold coated with 25 or 50 nm of gold target (Gold Target, dia. 57mm x 0.1mm, 81001, Cressington Scientific Instruments Ltd, UK) to be used as working and counter electrodes, respectively. The Cressington Sputter Coater (Cressington Sputter Coater 208HR, Cressington Scientific Instruments Ltd, UK) was used to coat the CFT squares, using a density and current settings of 19.3 gcm<sup>-3</sup> and 80 mA, respectively, and the AUTO/MTM mode using the Cressington Thickness Controller (MTM 20, Cressington Scientific Instruments Ltd, UK) to set the desired thickness. The 25 nm gold coated CFT electrodes were used as working electrodes (WEs; Figure 2.3B), while the 50 nm gold coated CFT had to be processed further to be used as counter electrodes (CEs).

#### 2.4.9.2 Counter electrode (CE) manufacturing

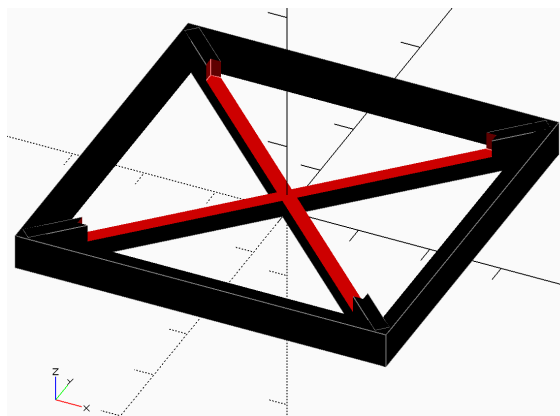
A 7 cm length of gold wire was cut. The wire was bent at 90°. The short section, 2 cm long, was inserted into the top-right of a 50 nm gold coated electrode and woven back and forth. A conductivity test (see Section 2.4.5) was done between the protruding gold wire tip (at the end



of the 5 cm section) and multiple spots on the electrode. An electrode was placed on top of a dry 6 cm<sup>2</sup> cellulose membrane square, prepared as the cellulose membrane (see Section 2.4.8). An epoxy frame was traced on the membrane square around the electrode. A second 6 cm<sup>2</sup> cellulose membrane square was placed on top and pressed down lightly along the epoxy frame to join both membranes. After the epoxy set, any excess membrane was cut off. The assembled isolated CE was visually inspected for any openings, which were sealed with epoxy when found. A photograph of an assembled CE can be seen in Figure 2.3C.

### 2.4.9.3 Electrode support and separation

An electrode support was 3D printed (see Section 2.4.3) to ensure the counter and working electrodes in half-cell A would not come into contact and to ensure that both would have adequate free diffusion of ions (see Figure 2.17 and Code B.5 for OpenSCAD source code).



**Figure 2.17** 3D printed electrode support. Top angled view of 3D printed electrode support to ensure the support of the working electrode and its separation from the counter electrode. It consists of a crossed frame (52 x 52 x 5 mm external measurements; frame thickness = 2 mm). A section (40 x 40 x 2 mm external measurements) was cut out of the centre to allow the working electrode to sit into the space. Ticks drawn every 10 mm.

### 2.4.10 Reference electrode manufacturing protocol

A silver/silver chloride (Ag/AgCl) reference electrode(RE) was made to carry out electrochemical measurements (Figure 2.4). Table 2.3 lists the components used for the reference electrode. The protocol developed to produce a RE is shown in Figure 2.4 and described below.

**Table 2.3** Specification of the components used to produce the reference electrode. The ‘Part’ column contains the name used to refer to the component throughout the text. BP, banana plug.

Part	Material	Brand	Part ID
Silver wire	Silver	Sigma-Aldrich	327026-4G
2 mm BP (white)	Gold plated brass	Multi Contact, DE	22.2618-29
Glass capillary	Glass (5 mm ext. dia. x 1 mm thick)	Fischer Labortechnik	2011005
Heat shrink	PTFE tubing (9.5 mm int. dia.)	Bohlender	BOHLS1828-40
Molecular sieve	$\text{K}_n\text{Na}_{12-n}[(\text{AlO}_2)_{12}(\text{SiO}_2)_{12}]\text{xH}_2\text{O}$ ; 4 Å, 3–5 mm beads	Alfa Aesar	L05359.30
NaOCl solution	Sodium hypochlorite (NaOCl); 10–14 % w/v available Cl	Scientific Laboratory Supplies	CHE-5362-S

#### **2.4.10.1 4 M KCl electrolyte solution**

29.82 g KCl were dissolved in 100 mL ultrapure water by stirring on a hot plate (endothermic reaction) and autoclaved (121 °C, 15 min).

#### **2.4.10.2 Glass capillary preparation**

A glass capillary tube was cut (7 cm length). A molecular sieve bead was placed on one end of the glass capillary. A 4 cm length of heat shrink was placed on the same end, with ca. 1 cm protruding from the capillary, thus encasing the molecular sieve. A gas burner was used to shrink the heat shrink until it became firmly attached (colour change to off-white). Care was taken to avoid breaking the heat shrink where it met the glass capillary. A second length of heat shrink (3 cm) was firmly attached at the top of the capillary with ca. 1 cm protruding. The capillaries were filled with ultrapure water using a long needle and syringe and left to stand. After 1 hr (min.), they were visually inspected for leaks. The water was discarded using syringe and needle.

#### **2.4.10.3 Silver wire preparation**

A 7 cm length of silver (Ag) wire was cut and soldered onto a 2 mm BP. To deposit AgCl, the Ag wire was incubated in a container filled with NaOCl (at a height where the solder was minimum 0.5 cm above the solution) for 30 min at room temperature. This step was repeated after washing the Ag wire with ultrapure water. If needed, the Ag/AgCl wire was stored in 4 M KCl.

#### **2.4.10.4 Reference electrode assembly**

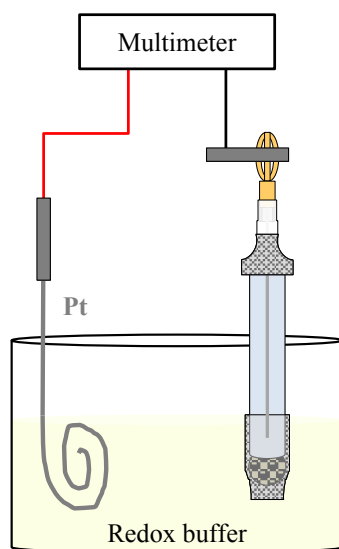
The capillary was washed with 4 M KCl and then filled again with 4 M KCl leaving about 0.5 cm of headspace. The Ag/AgCl wire was inserted into the BP insulator and then into the capillary tube. The insulator sat on the edge of the capillary. A gas burner was used to fix the insulator in place with the heat shrink. Care was taken not to heat the electrolyte solution (4 M KCl) to avoid salt precipitation due to water evaporation. The reference electrode (RE) was stored upright at room temperature with the molecular sieve tip submerged in 4 M KCl. The voltage of the REs was determined as described in the “Reference electrode characterisation” section below. The reference electrodes were degassed by being placed in the anaerobic chamber (MACS-MG-500 anaerobic workstation, Don Whitley Scientific, Shipley, UK) for 3 days prior to their use.

#### **2.4.11 Reference electrode characterisation**

After production, the REs were stored in 4 M KCl overnight (min.) to ensure the hydration of the molecular sieve and, thus, the conductivity between the Ag/AgCl of the RE and the buffer used for its characterisation. The potential of the RE ( $V_{RE}$ ) v SHE was determined using a handheld

digital multimeter (Fluke-115, Fluke, UK). The test probe tips were replaced by crocodile clips. The positive (red) probe was connected to a platinum (Pt) wire (0.127 mm diameter, #00263, Alfa Aesar, UK). The negative (black) probe was connected to the banana plug on the RE. Both the Pt wire and RE were introduced into the redox buffer (51350060, Mettler Toledo, UK) and the DC voltage was measured ( $V_m$ ). The measured voltage (in V) was subtracted from 0.427 V (Ag/AgCl potential of the redox buffer v SHE at 20 °C) as shown in Equation 2.1. See Figure 2.5 for a visual schematic of how the two potential reference scales are mapped.

$$x = 0.427 - V_m \quad (2.1)$$



**Figure 2.18** Electrochemical set up required to characterise the reference electrodes (RE) and calculate the value to translate potentials across SHE and Ag/AgCl reference scales. Crocodile clips (grey) were fitted on the leads (red and black) of a hand-held digital multimeter (Fluke-115, Fluke, UK). A platinum (pt) wire (left) and a reference electrode (right) were connected to the multimeter and the DC voltage was measured.

#### 2.4.12 Production of the *modified stopper*

The CE and WEs within the electrochemical cell had to be connected to the potentiostat. Rubber stoppers were modified to enable gold wires (7 cm long) to pass from the interior to the exterior of the cell using 3D printed objects, a holder and support, designed to help pierce the rubber stopper along the centre (see Code B.3 and B.4). Each gold wire was soldered to a 2 mm BP (used to connect to BSs attached to the container wall connectors; see Section 2.4.16), which was then fixed on the rubber stopper with epoxy. The resulting assembly is referred to as “*modified stopper*” and Figure 2.7 shows the steps followed to produce them.

#### 2.4.13 Electrochemical cell assembly

The electrochemical cell was assembled by placing the working and counter electrodes on the corresponding half-cells. See Figure 2.6 for a representation viewed from the inside and Figure 2.2 for the port labels. The half-cells were joined once they contained the working and counter

electrodes, with the rubber gaskets containing the membrane placed between them. Figure 2.2 shows the labels used to refer to the half-cell ports. These steps are described below in more detail.

#### **2.4.13.1 CE assembly**

A *modified stopper* (see Section 2.4.12) was introduced into port A3 of a clean half-cell A (see Figure 2.2). The CE was placed into the opening of half-cell A and the protruding wire was twisted together with the wire from the *modified stopper*. These were soldered together by brushing a small amount of solder (see Section 2.4.4). A conductivity test (see Section 2.4.5) was done between the gold wire protruding from the CE and the 2 mm BP on the *modified stopper*. Once the CE was in place, the electrode support (see Section 2.4.9.3) was placed within the half-cell A opening, making sure the gold wire was tucked behind the support or up in the opening to prevent short circuiting between CE and W2.

#### **2.4.13.2 WE assembly**

*Modified stoppers* (see Section 2.4.12) were inserted into port B2 of a clean half-cell B and into port A2 of the half-cell A used in the previous point (containing the CE and electrode support; Section 2.4.13.1). Ca. 2 cm of the end of the gold wire from a *modified stopper* were bent at 90° and inserted into the top centre of a WE and weaved into the WE multiple times. The WE was then pushed into the half-cell opening. A conductivity test (see Section 2.4.5) was done between multiple points on each WE and the 2 mm BP on the *modified stopper* to which it was connected.

#### **2.4.13.3 Gasket and membrane ‘sandwich’ assembly**

Just prior to assembling the half-cells, a membrane was ‘sandwiched’ between two rubber gaskets. High vacuum grease (Dow Corning, UK) was applied to the edge of the inner square of two rubber gaskets with a toothpick. A flat membrane was placed on one of the rubber gaskets making sure the inner square was well covered. The second gasket was placed on top of the membrane, with the vacuum grease facing the membrane. Sharp tweezers were heated using a gas burner and used to pierce the membrane where the bolt holes were located (four bolt holes total).

#### **2.4.13.4 Electrochemical cell assembly**

Figure 2.19 contains a schematic representation of the electrochemical cell assembly. Half-cell A, containing the CE, electrode support and WE (henceforward referred to as W2), was laid with the opening facing up. Four hex bolts were fitted with a small and then a large washer. Half-cell

A was lifted and the hex bolts were inserted into the bolt holes, making sure the large washer was in contact with the half-cell. Next, the gasket and membrane ‘sandwich’ (see Section 2.4.13.3) was inserted into the bolts and placed on half-cell A, making sure of its orientation prior to the insertion as the bottom frame of both the cell and gasket is larger than the top (see Figure 2.2). Half-cell B containing a WE (henceforward referred to as W1) was then placed on top of half-cell A, making sure W1 was not trapped between them. Finally, a large washer, a small washer and a hex nut were placed on each bolt and the nuts were hand-tighten in preparation of cell sterilisation (Section 2.4.14). For immediate use, the nuts were tighten using adjustable spanners. Note that ports B1 and B3 were open at this point (i.e. stoppers had not been introduced).

#### **2.4.14 Electrochemical cell sterilisation and degassing**

The hand-tighten bolts were loosen to allow for metal expansion during autoclaving (121 °C, 15 min) using a desktop autoclave (ST 19 T, Dixon, Wickford, UK). The cell was covered with foil prior to their introduction into the autoclave. No. 7 and No. 9 stoppers were wrapped in foil to be autoclaved separately. The electrochemical cell was left to cool completely after autoclaving. In a laminar flow hood, all the bolts were tightened using two adjustable spanners. The foil was replaced to ensure its continued sterility. The cells were transferred into an anaerobic chamber (MACS-MG-500 anaerobic workstation, Don Whitley Scientific, Shipley, UK) and left to degas for 6 days minimum. The RE was degassed separately and introduced into the B3 port of the degassed electrochemical cell when required.

#### **2.4.15 Anaerobic container preparation**

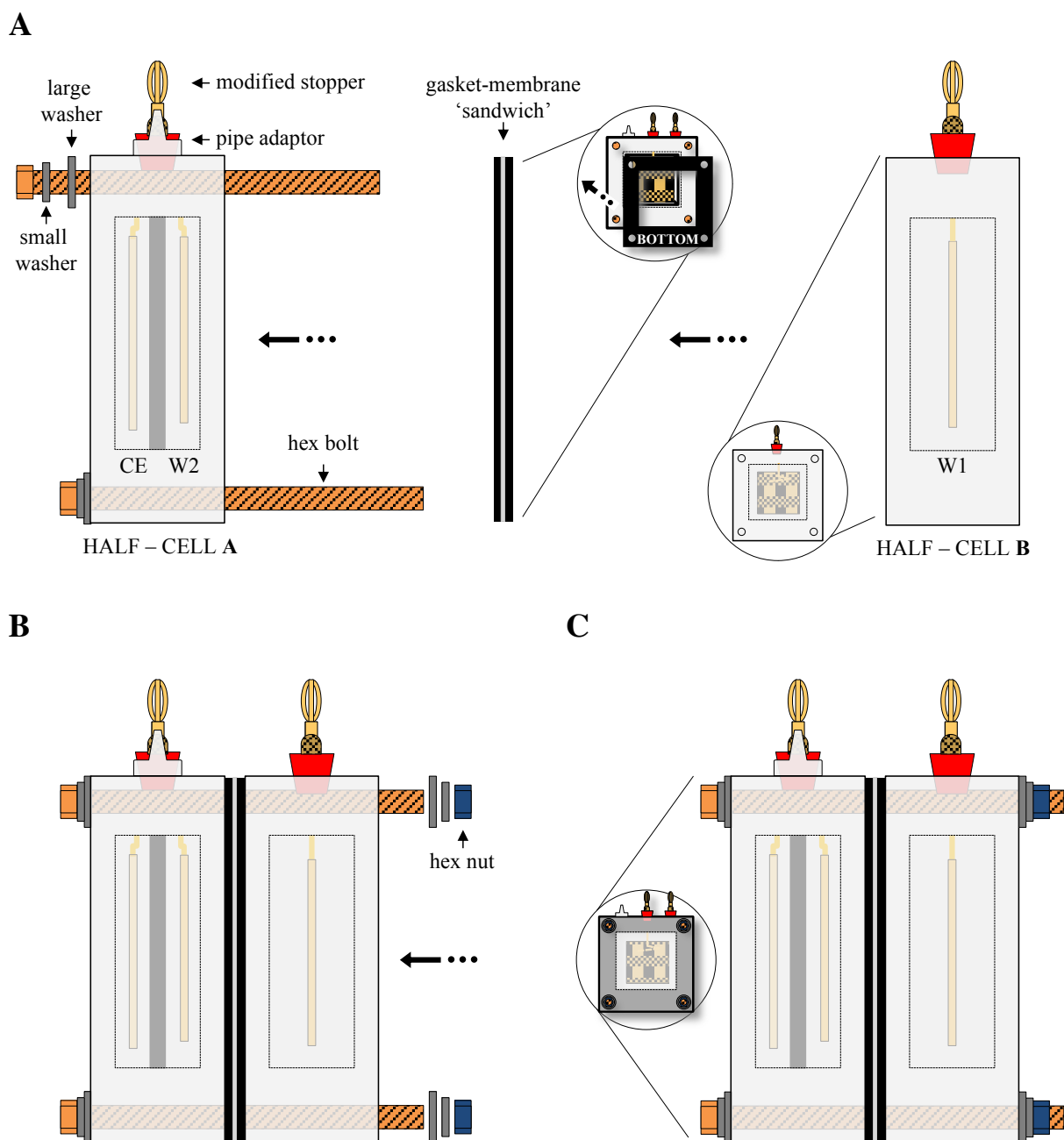
Plastic containers (9 Litre Rectangular Container with clip lid, Lock & Lock, UK) were modified to enable electrical connections from the electrochemical cells to the potentiostat. Table 2.4 contains the component specification for the electronic connections.

##### **2.4.15.1 Container anaerobicity test**

A sterile Erlenmeyer flask with a plug covered with foil was placed in the anaerobic chamber for a minimum of 12 h. Once degassed, it was filled with sterile anaerobic medium (see Chapter 3, Section 3.4.2) and transferred to the container. The container was closed and removed from the anaerobic chamber and visually monitored for changes in the media colour.

##### **2.4.15.2 External container connections**

Holes were drilled onto two sides of the containers to place banana connectors, as shown in Figure 2.9. The steps followed to screw the 4 mm BSs and to place the insulators of four white



**Figure 2.19** Schematic representation of the electrochemical cell assembly viewed from SIDE A (see Figure 2.2). **A** Four hex bolts containing small and large washers were inserted into the assembled half-cell A (i.e. containing CE and W2). The gasket-membrane 'sandwich' (see Section 2.4.13.3) was inserted into the bolts, making sure it was correctly aligned, and placed on half-cell A. Half-cell B (with W2 previously placed) was inserted and placed on top of the gasket-membrane 'sandwich'. **B** Large and small washers, followed by hex nuts, were inserted into the four bolts and tightened (by hand or using adjustable spanners). **C** The assembled electrochemical cell. The circled areas show the view from the front, rotated clockwise on the z-axis.

**Table 2.4** Specification of the electronic components used to connect the Gamry potentiostat to the electrochemical cells. The ‘Part’ column contains the name used to refer to the component throughout the text. BP, banana plug; BS, banana socket.

Part	Material	Brand	Part ID
2 mm BS (black)	Gold plated brass	Multi Contact, DE	22.2360-21 22.1038
2 mm BS (red)	Gold plated brass	Multi Contact, DE	22.2360-22 22.1038
2 mm BP (black)	Gold plated brass	Multi Contact, DE	22.2618-21
2 mm BP (brown)	Gold plated brass	Multi Contact, DE	22.2618-27
2 mm BP (white)	Gold plated brass	Multi Contact, DE	22.2618-29
4 mm BS (brown)	Gold plated brass	Hirschmann Test & Measurement, DE	972354105
4 mm BS (violet)	Gold plated brass	Hirschmann Test & Measurement, DE	972354109
4 mm BS (grey)	Gold plated brass	Hirschmann Test & Measurement, DE	972354106
4 mm BS (black)	Gold plated brass	Hirschmann Test & Measurement, DE	972354100
4 mm BP	Gold plated brass	Multi Contact, DE	22.2380-25 22.1203
Gold wire	Gold wire, >99.9% (0.25mm dia.)	Alfa Aesar	010967.BY
Ag-Cu wire	Silver plated copper (PTFE insulation)	Alpha Wire, UK	5855 BK005
Copper tape	Copper foil (conductive acrylic adhesive coating)	Advance Tapes, UK	542-5511

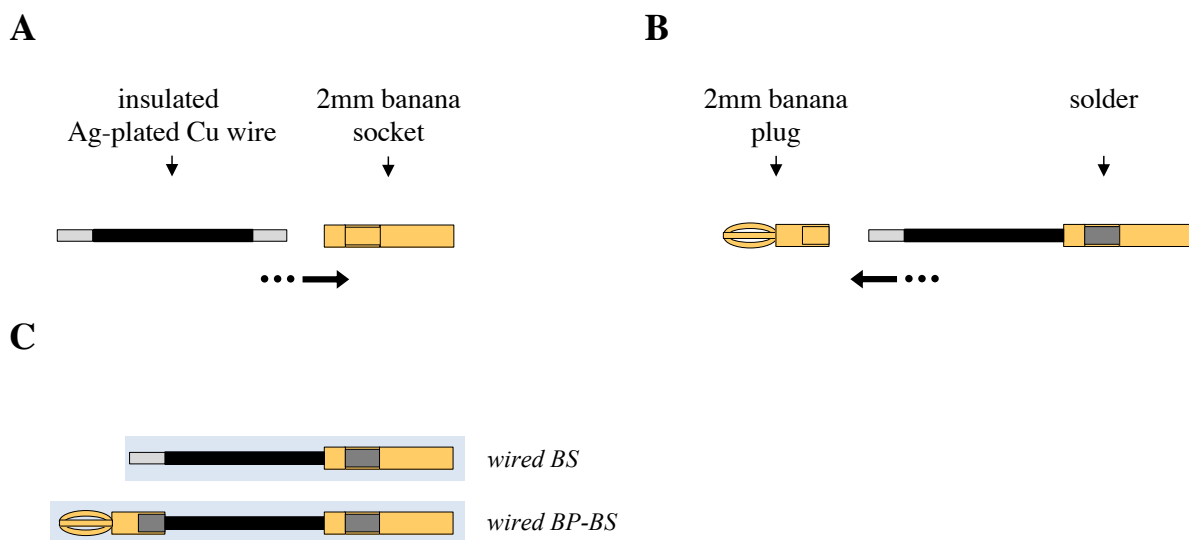
2 mm BPs onto the corresponding positions on the container wall are described in Figure 2.10 (see Table 2.4 for component information). All components were fixed in place using epoxy (see Section 2.4.2).

### 2.4.15.3 Internal container connections

Two types of internal connections were used. Insulated silver plated copper (Ag–Cu) wires soldered at 325 °C onto 2 mm BSs (referred to as “*wired BSs*”) were connected to the screws of the 4 mm BSs (Figure 2.20A and B). 2 mm BP were soldered onto the *wired BSs*, referred to as *wired BP–BSs*, (Figure 2.20C and D) to connect the REF Gamry lead (see Table 2.1), by placing the BP through the insulation fitted on the container wall. The soldered areas of both the *wired BSs* and *wired BP–BSs* were incubated in 96% ethanol (VWR Chemicals, FR) to remove the flux components and prevent oxidation over time. Figure 2.11 illustrates how these components were connected to the container wall.

### 2.4.15.4 Electrical noise elimination

Copper tape (542-5511, Advance Tapes, UK) was placed in both the main container and its lid as shown in Figure 2.13. To make a pseudo Faraday cage, copper tape was placed along the inner sides and edges of the container and an ‘X’ was made at the bottom and on the lid. A length of silver plated copper (Ag–Cu) wire was screwed onto a black BS on the container wall. The other end was fixed onto the container wall by putting copper tape on it and then hot glue where the copper tape, the wire and the container met (Figure 2.10; hot glue not shown). A *wired BPs* (Ag–



**Figure 2.20** Assembly of the internal connection components *wired BS* and *wired BP-BS*. **A** The silver plated copper (Ag–Cu) wire was soldered into the 2 mm BS, producing a *wired BS*. **B** A 2 mm banana plug is soldered onto the other end of the Ag–Cu wire, producing a *wired BP-BS*. **C** The manufactured *wired BS* (top) and a *wired BP-BS* (bottom).

Cu wire soldered onto a 2 mm BP) was screwed onto a black BS on the container wall as shown in Figures 2.13D and Figure 2.10. Finally, a *wired BSs* was placed on the lid of the container with copper tape on top and hot glued to prevent its coming loose (Figure 2.13E). All containers were placed inside a Faraday cage (M600600350GE, Hoffman, nVent, USA).

The containers were grounded with the use of a *GND BP*. A *GND BP* refers to a gold plated BP with a Ag–Cu wire, connected to the Faraday cage by using copper tape to hold the wire in place. The *GND BP* was then placed on one of the black BS in the container. A conductivity test (Section 2.4.5) was carried out from the Faraday cage to multiple points on the copper tape on the lid as well as other black BSs.

#### 2.4.16 Connections required for ZRA measurements

The connections required to measure the electrochemical cells in ZRA mode are shown in Table 2.5.

#### 2.4.17 Experimental platform set-up

The degassed electrochemical cells were be fitted with degassed reference electrodes in either port B1 or B3 (Figure 2.2) within the anaerobic chamber. The opening was closed with the addition of PTFE tape around the capillary were the latter met the cell. After being filled with medium and inoculated in the anaerobic chamber as required (biotic elements described in Chapter 3, Section 3.4.5), open ports were closed with the aid of rubber stoppers (No. 9 for port B1 or No. 7 for port B3). A gas collection bag was connected to port A1 using tubing (Table 2.2 for component



**Table 2.5** Electronic connections from Gamry to electrochemical cell electrodes for ZRA measurements

Gamry cable colour	Name	Container wall connection colour (type)	Electrochemical cell electrode
Blue	Working Sense (WS)	Grey (4 mm BS)	W1
Green	Working Electrode (W)	Grey (4 mm BS)	W1
White	Reference (REF)	White (2 mm BP)	RE
Red	Counter Electrode (C)	Purple (4 mm BS)	W2
Orange	Counter Sense (CS)	Purple (4 mm BS)	W2
Long Black	Floating Ground (GND)	Black (4 mm BS)	–
Short Black	Chassis (Earth) Ground (GND)	Black (4 mm BS)	–
–	–	Brown (4 mm BS) (x 2)	CE

BP, banana plug; BS, banana socket. Note that the CE is not connected during ZRA measurements. *Table modified from Gamry ECM8's manual*

details). Three chemical anaerobic atmosphere generation sachets (Anaerogen® pack, AN0035, Oxoid, Thermo Scientific, UK) were introduced into the container to catalyse any oxygen that permeated into the container. The banana connector of the lid was connected to the corresponding connector in the container. The container was then closed and removed from the anaerobic chamber. Each container was grounded by connecting it with a *GND BP* (see Section 2.4.15.4). The containers were placed in the Faraday cage and connected to the potentiostat using the MUX channel leads as shown in Table 2.5. A schematic representation of the complete system is shown in Figure 2.1.

## Chapter 3

# Towards investigating the “syntrophy over wires” hypothesis

### 3.1 Introduction

In the Introduction chapter, the thermodynamic “world-view” of metabolism was discussed in Section 1.2. The metabolic half reaction in the reduction direction can be organised based on their reduction potential. It can be considered that an organism’s metabolism is limited by the reactions it can catalyse as determined by their genes and by the energetics of the reactions, which are also dependent on the environment characteristics, mainly on compound concentrations, pH and temperature (Figure 1.7). Some metabolic interactions lead to syntrophic growth when the interaction is beneficial and required for all species involved.

One example of these syntrophic interactions is the coculture between *Desulfovibrio vulgaris* (Dv) and *Methanococcus maripaludis* (Mm), referred here as DvMm. As described in Section 1.7.1, Dv is a sulphate reducer, while Mm is a methanogenic archaeon. Both microorganisms are strict anaerobes that are normally grown at 37 °C. The basis of the syntrophy is represented in Figure 1.7. In the presence of sulphate, Dv consumes lactate and reduces sulphate to produce acetate and CO<sub>2</sub> (Noguera et al., 1998). This is a multi-step reaction and pyruvate is one of the intermediaries. The reduction from pyruvate to acetate is the final step and it is the one represented in Figure 1.7. Under these conditions, H<sub>2</sub> is not accumulated and, hence, Mm does not have a source of electrons to reduce CO<sub>2</sub> to CH<sub>4</sub>. However, in the absence of sulphate, Dv cannot carry out its final reduction steps and H<sub>2</sub> is produced as well as acetate, thereby providing electrons for Mm to catalyse the CO<sub>2</sub> reduction. The syntrophy is based on the fact that Dv suffers from H<sub>2</sub> thermodynamic inhibition, so its growth is limited. Therefore, not only is Mm able to grow, but by so doing, it removes H<sub>2</sub> from the environment, alleviating Dv’s inhibition, and enabling its growth.

Because DvMm’s coculture is based on H<sub>2</sub> exchanged, it was hypothesised that the syntro-

phy could also be established by substituting the  $H_2$  with electrodes in a bioelectrochemical system (BES). Therefore, Dv could use an electrode as an electron sink, while Mm could use the electrode as an electron source. This is referred to as the “syntrophy over wires” hypothesis throughout this work.

There are three main ways by which organisms interact with electrodes: (i) direct electron transfer by membrane-bound enzymes, (ii) direct long-range electron transfer (e.g. conductive pili), and (iii) indirect electron transfer facilitated by the use of soluble electron carriers or shuttles, both organic and inorganic, as mentioned in Chapter 1. Dv and Mm have both been grown as monocultures on electrodes, as shown in the work of Croese et al. (2011) and Lohner et al. (2014), respectively. However, the mechanisms by which these microorganisms interact with the electrodes is not thoroughly understood. Below is an overview of what has been found or proposed in the literature. Fortunately, both Dv and Mm’s genomes have been sequenced by Heidelberg et al. (2004) (RefSeq, DVU) and Hendrickson et al. (2004) (GenBank, MMP), respectively, facilitating homology searches and gene annotation queries. All the information retrieved has been summarised in Figure 3.1 and Tables 3.1 and 3.2 for Dv and Mm, respectively.

It has been suggested that both Dv and Mm interact by means of indirect electron transfer by utilising  $H_2$  as an inorganic electron carrier as both organisms carry out hydrogen metabolism.

**Table 3.1** Summary of Dv’s enzymes potentially involved in electron transfer.

Protein	Gene	Gene_ID	Source
Coo	<i>cooM</i>	DVU2286	Walker et al. (2009)
	<i>cooK</i>	DVU2287	Walker et al. (2009)
	<i>cooL</i>	DVU2288	Walker et al. (2009)
	<i>cooX</i>	DVU2289	Walker et al. (2009)
	<i>cooU</i>	DVU2290	Walker et al. (2009)
	<i>cooH</i>	DVU2291	Walker et al. (2009)
	<i>cooF</i>	DVU2293	Walker et al. (2009)
	<i>cooS</i>	DVU2098	Walker et al. (2009)
Hmc	<i>hmc</i>	DVU0531	Walker et al. (2009)
	<i>hmc</i>	DVU0532	Walker et al. (2009)
	<i>hmc</i>	DVU0533	Walker et al. (2009)
	<i>hmc</i>	DVU0534	Walker et al. (2009)
	<i>hmc</i>	DVU0535	Walker et al. (2009)
	<i>hmc</i>	DVU0536	Walker et al. (2009)
Hyn	<i>hynB-1</i>	DVU1921	Walker et al. (2009)
	<i>hynA-1</i>	DVU1922	Walker et al. (2009)
Hyd	<i>hydA</i>	DVU1769	Walker et al. (2009)
	<i>hydB</i>	DVU1770	Walker et al. (2009)
Ccm	<i>ccmC</i>	DVU1047	Croese et al. (2011)
	<i>ccmB</i>	DVU1048	Croese et al. (2011)

Protein	Gene	Gene_ID	Source
	<i>ccmF</i>	DVU1050	Croese et al. (2011)
	<i>ccmE</i>	DVU1051	Croese et al. (2011)
Flg	<i>flgE</i>	DVU0307	Croese et al. (2011)
	<i>flgC</i>	DVU0315	Croese et al. (2011)
	<i>flgB</i>	DVU0316	Croese et al. (2011)
	<i>flgG</i>	DVU0512	Croese et al. (2011)
	<i>flgG</i>	DVU0513	Croese et al. (2011)
	<i>flgA</i>	DVU0514	Croese et al. (2011)
	<i>flgH</i>	DVU0515	Croese et al. (2011)
	<i>flgI</i>	DVU0516	Croese et al. (2011)
	<i>flgK</i>	DVU0519	Croese et al. (2011)
	<i>flgL</i>	DVU0520	Croese et al. (2011)
	<i>flgM</i>	DVU0523	Croese et al. (2011)
	<i>flgE</i>	DVU1443	Croese et al. (2011)
	<i>flgD</i>	DVU1444	Croese et al. (2011)
	<i>flgC</i>	DVU2893	Croese et al. (2011)
Fdh	<i>fdhE</i>	DVU0577	Deutzmann et al. (2015)
	<i>fdhD</i>	DVU0578	Deutzmann et al. (2015)
	<i>fdhE</i>	DVU2810	Deutzmann et al. (2015)

Walker et al. (2009) proposed a mechanism for syntrophic growth between Dv and Mm, where the hydrogenases expressed by Dv (*coo*, *hmc*, *hyn* and *hyd*) played a crucial role. Lie et al. (2012) described the importance of the hydrogenase Eha (energy-conserving hydrogenase A) in electron flow and energy conservation in methanogenic Archaea. Additionally, Costa et al. (2013) demonstrated that the hydrogenase *fru*, *frc*, *hmd*, *vhu*, *vhc* and *ehb* were required for Mm to grow on H<sub>2</sub> and CO<sub>2</sub>.

Furthermore, multiple studies have suggested alternative indirect electron transfer mechanisms for both organisms. Deutzmann et al. (2015) proposed that redox-active enzymes, such as hydrogenases and formate dehydrogenase (Fdh), can be released from the cell, adsorb to, and subsequently interact with an electrode surface. Both Dv's and Mm's genomes have been annotated with Fdh in addition to the hydrogenases mentioned above.

Croese et al. (2011) found that Dv's genome shows c-type cytochrome genes with homology to the proteins involved in electron transfer in *Geobacter* species. As cytochrome c exists both in soluble and membrane-bound forms (Yeagle, 2016), both direct or indirect electron transfer could take place. Figure 3.1 includes this speculation by including question marks. A search through

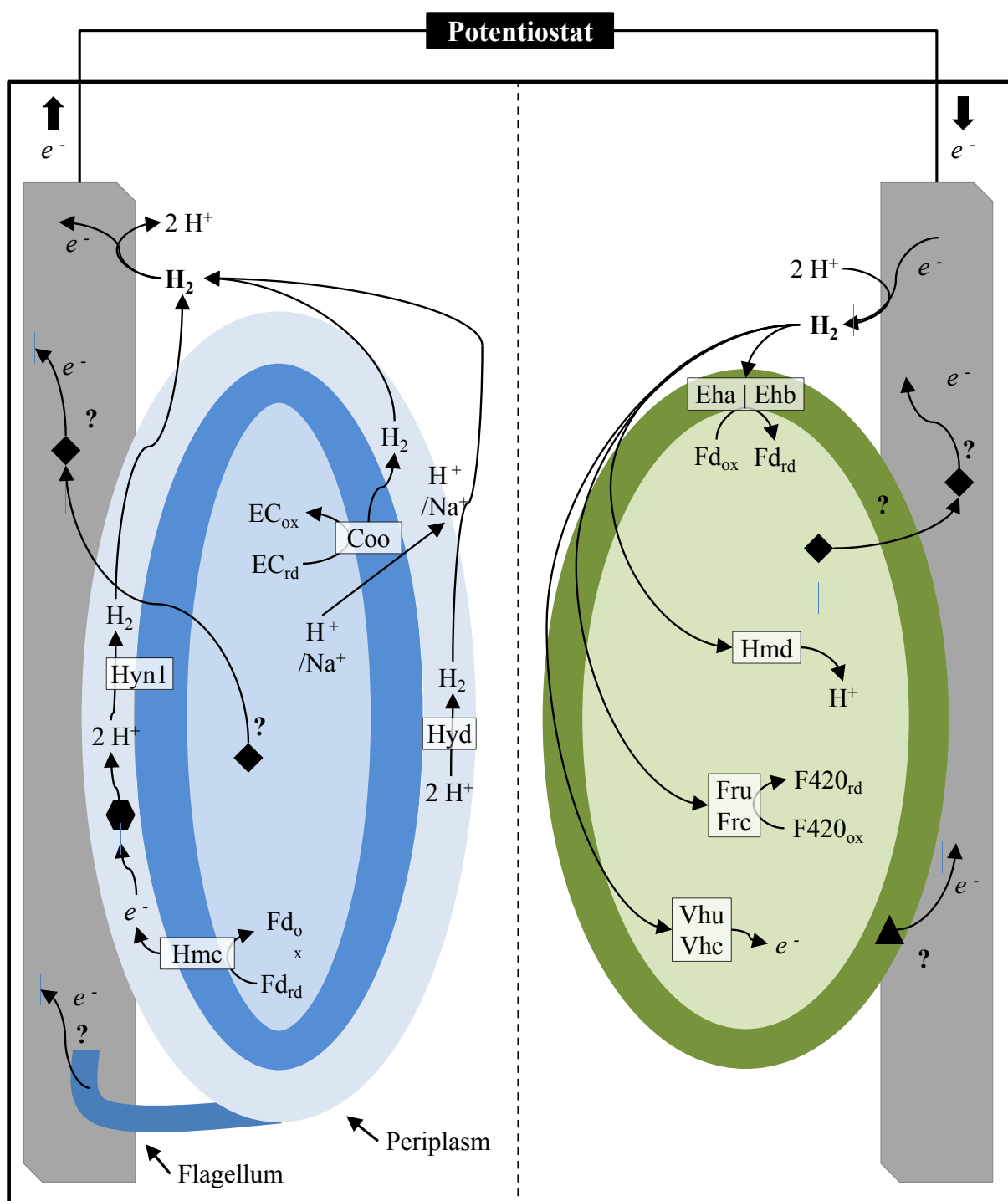
**Table 3.2** Summary of Mm's enzymes potentially involved in electron transfer.

Protein	Gene	Gene_ID	Source
Fru	<i>fruA</i>	MMP1382	(Costa et al., 2013)
	<i>fruD</i>	MMP1383	(Costa et al., 2013)
	<i>fruG</i>	MMP1384	(Costa et al., 2013)
	<i>fruB</i>	MMP1385	(Costa et al., 2013)
Frc	<i>frcB</i>	MMP0817	(Costa et al., 2013)
	<i>frcG</i>	MMP0818	(Costa et al., 2013)
	<i>frcD</i>	MMP0819	(Costa et al., 2013)
	<i>frcA</i>	MMP0820	(Costa et al., 2013)
Hmd	<i>hmd</i>	MMP0127	(Costa et al., 2013)
Vhu	<i>vhuA</i>	MMP1694	(Costa et al., 2013)
	<i>vhuB</i>	MMP1692	(Costa et al., 2013)
	<i>vhuG</i>	MMP1695	(Costa et al., 2013)
	<i>vhuD</i>	MMP1696	(Costa et al., 2013)
	<i>vhuU</i>	MMP1693	(Costa et al., 2013)
Vhc	<i>vhcA</i>	MMP0823	(Costa et al., 2013)
	<i>vhcG</i>	MMP0822	(Costa et al., 2013)
	<i>vhcD</i>	MMP0821	(Costa et al., 2013)
Ehb	<i>ehbQ</i>	MMP0400	(Costa et al., 2013)
	<i>ehbP</i>	MMP0940	(Costa et al., 2013)
	<i>ehbB</i>	MMP1049	(Costa et al., 2013)
	<i>ehbC</i>	MMP1073	(Costa et al., 2013)
	<i>ehbD</i>	MMP1074	(Costa et al., 2013)
	<i>ehbN</i>	MMP1153	(Costa et al., 2013)
	<i>ehbA</i>	MMP1469	(Costa et al., 2013)
	<i>ehbO</i>	MMP1621	(Costa et al., 2013)
Eha	<i>ehbM</i>	MMP1622	(Costa et al., 2013)
	<i>ehbL</i>	MMP1623	(Costa et al., 2013)
	<i>ehbK</i>	MMP1624	(Costa et al., 2013)
	<i>ehbJ</i>	MMP1625	(Costa et al., 2013)
	<i>ehbG</i>	MMP1627	(Costa et al., 2013)
	<i>ehbF</i>	MMP1628	(Costa et al., 2013)
	<i>ehbE</i>	MMP1629	(Costa et al., 2013)
	<i>ehaA</i>	MMP1448	Lie et al. (2012)
	<i>ehaB</i>	MMP1449	Lie et al. (2012)
	<i>ehaC</i>	MMP1450	Lie et al. (2012)
	<i>ehaD</i>	MMP1451	Lie et al. (2012)
	<i>ehaE</i>	MMP1452	Lie et al. (2012)
	<i>ehaF</i>	MMP1453	Lie et al. (2012)
	<i>ehaG</i>	MMP1454	Lie et al. (2012)
	<i>ehaH</i>	MMP1455	Lie et al. (2012)
	<i>ehaI</i>	MMP1456	Lie et al. (2012)
	<i>ehaJ</i>	MMP1457	Lie et al. (2012)
	<i>ehaK</i>	MMP1458	Lie et al. (2012)
	<i>ehaL</i>	MMP1459	Lie et al. (2012)
	<i>ehaM</i>	MMP1460	Lie et al. (2012)
	<i>ehaN</i>	MMP1461	Lie et al. (2012)
	<i>ehaO</i>	MMP1462	Lie et al. (2012)
	<i>ehaP</i>	MMP1463	Lie et al. (2012)
	<i>cycZ</i>	MMP0957	Lie et al. (2012)

Dv’s genome annotations identified a cytochrome c-type biogenesis protein (Ccm), cytochrome c oxidase (Cox) and cytochrome c-553 . As the second involves a reaction with molecular oxygen and the third is involved in sulphur metabolism, only the first was included in the summary table. A search through Mm’s genome annotations also found a cytochrome c-type biogenesis protein (Cyc), which has also been included in the summary table (Table 3.2).

Croese et al. (2011) proposed that the flagellar genes associated with physical association during syntrophic growth (Walker et al., 2009) could be involved in the adherence to the electrodes and could also potentially be involved in electron transfer. Finally, Lohner et al. (2014) showed Mm’s uptake and metabolism of electrons in a hydrogenase-independent manner, but the mechanism(s) remain(s) unknown. Again, the speculation of electron transfer through the flagellum and in alternative means has been indicated by a question marks in Figure 3.1.

To investigate the “syntrophy over wires” hypothesis, an electrochemical platform was designed and produced, which would enable experiments to be carried out under strict anaerobic conditions with a large experimental design (see Section 3.4.1). Briefly, an electrochemical cell was designed to have two compartments (called “half-cells”), separated by a permeable membrane. Each half-cell contained a working electrode (WE), referred to as W1 and W2. Each microorganism was inoculated into a different half-cell. The growth of both organisms was expected by an exchange of electrons through connecting the two working electrodes (WEs, i.e. W1 and W2) together (i.e. short-circuiting), which was evaluated by monitoring the current between them in zero resistance ammeter (ZRA) mode. Additional electrochemical techniques were used to characterise the system. Electrochemical impedance spectroscopy (EIS) was used to characterise the formation of biofilms on the working electrodes, while cyclic voltammetry (CV) was used to investigate the electron transfer reactions present (Marsili et al., 2008).



**Figure 3.1** Potential mechanisms involved in the interaction between Dv and Mm and the electrodes. Both Dv (left) and Mm (right) employ indirect electron transfer mediated by  $H_2$ . Dv produces  $H_2$  through four different hydrogenases (Coo, Hmc, Hyd and Hyn), which permeates out of the cell. The periplasmic cytochrome  $c_3$  (black hexagon) plays a role in providing the electrons for  $H_2$  production to some enzymes. If this protein were released from the cell, it could theoretically transfer electrons to the electrode. C-type cytochromes and flagellar genes similar to those involved in EET in *Geobacter* species have been identified in Dv's genome, indicating possible alternative electron transfer mechanisms. Mm utilises  $H_2$  with the hydrogenases Frc, Fru, Vhc, Vhu, Eha and Ehb. Hydrogenase-independent electron transfer has been reported for Mm although the mechanism(s) remain(s) unknown. Possible mechanisms include direct and indirect electron transfer. ▲, membrane-bound molecules involved in direct electron transfer. ◆, organic molecules involved in indirect electron transfer. Data from Costa et al. (2013); Croese et al. (2011); Goyal et al. (2016); Lie et al. (2012) and Walker et al. (2009).

## 3.2 Results

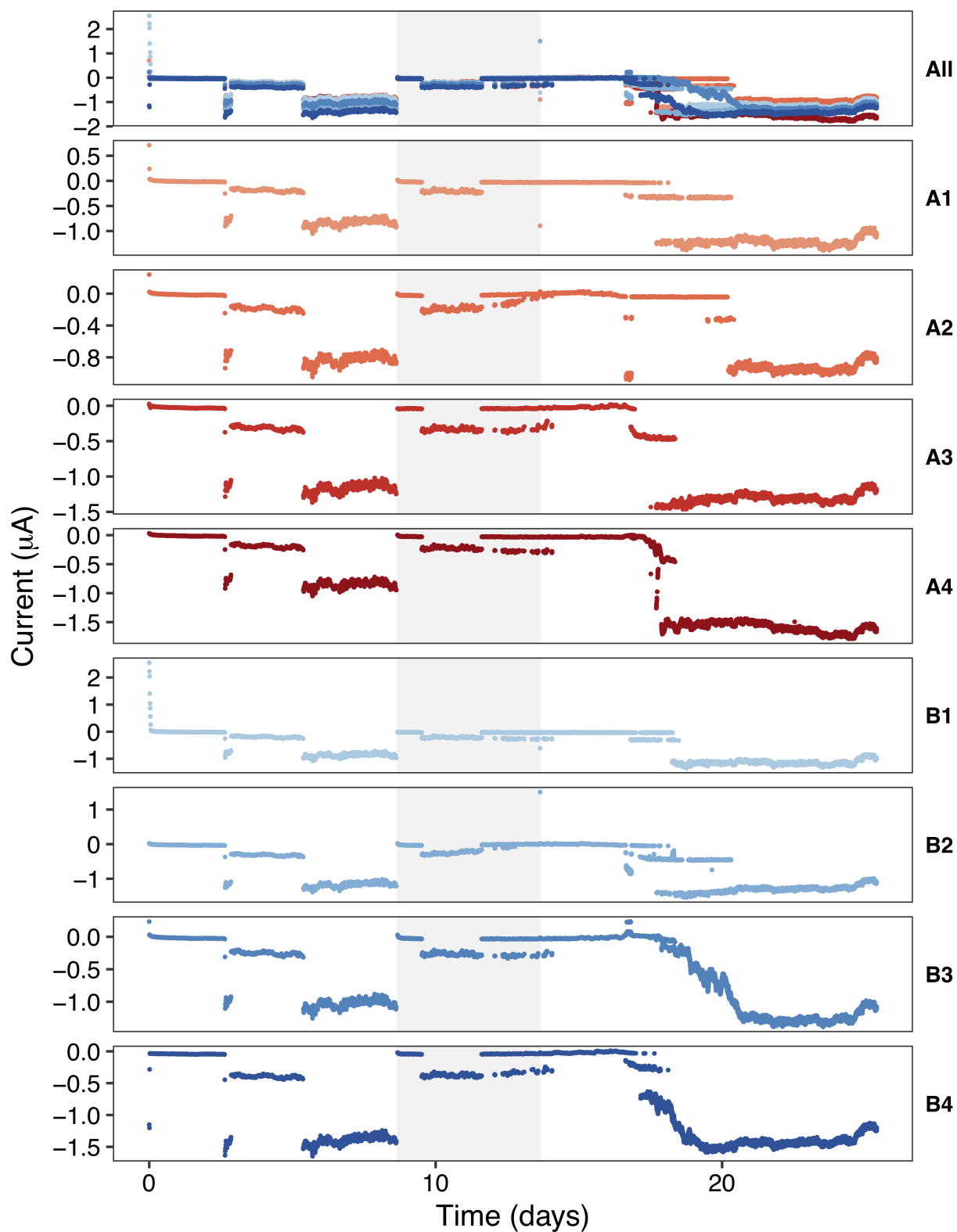
In this section, the first attempt to investigate the novel configuration described as the “syntrophy over wires” hypothesis in Chapter 1, Section 1.7.1 and in the Introduction to this chapter has been described. Twelve electrochemical cells were prepared to this purpose (Section 3.4.1). Eight cells were inoculated with Dv in half-cell B (HC B) containing the working electrode 1 (W1) and Mm in half-cell A (HC A), which contains the working electrode 2 (W2). Four of these cell, referred to as the biotic (B) condition, were connected to the potentiostat in zero resistance ammeter (ZRA) mode (see Section 3.4.8), thus shorting W1 and W2. The other four cells were not monitored (i.e. they were not externally connected to the potentiostat) and thus the circuit between W1 and W2 was left open. Therefore, these cells are referred to as being the non-connected biotic (nB) condition. The remaining four cells were not inoculated and are hence referred to as the abiotic (A) condition. The abiotic cells were also connected to the potentiostat in ZRA mode. The current between W1 and W2 was monitored for 25 days for both the biotic and abiotic conditions.

### 3.2.1 Current

The expectation under the “syntrophy over wires” hypothesis is that Dv and Mm are able to grow syntrophically in the biotic condition, because Dv could use W1 as an electron sink after lactate consumption, while Mm would use W2 as an electron source to reduce  $\text{CO}_2$  to  $\text{CH}_4$ . In other words, Dv would provide Mm with the required electrons for methanogenesis. As such, a positive current to be generated throughout the experiment was expected. Furthermore, the current would be proportional to the biofilm size of both electrodes and, as such, mimic the shape of a typical bacterial growth curve (Marsili et al., 2008).

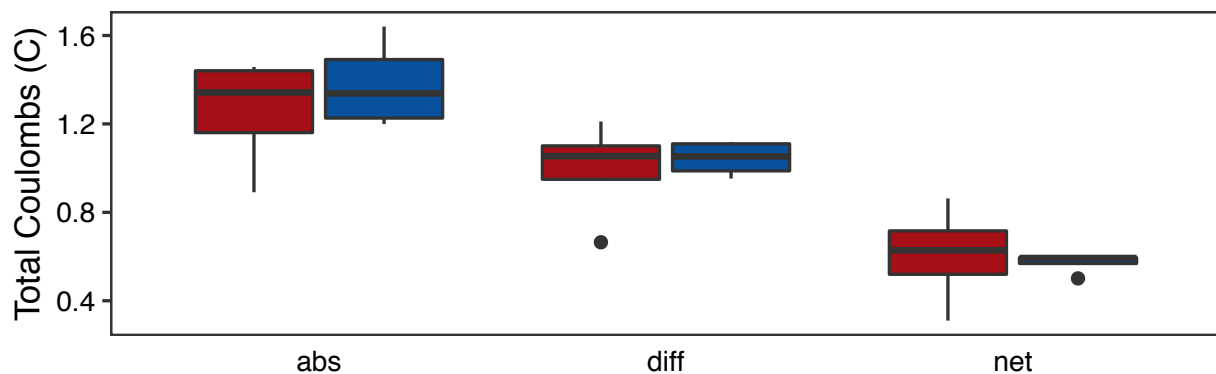
The current between the cells working electrodes (W1 and W2) was recorded throughout the experiment in three stages as can be observed in Figure 3.2. The first 10 time points of the data were excluded for the first and second stages because of the high measurements recorded due to the circuit discharge when shorted. The complete data can be seen in Figure D.1. The second stage period is indicated by a grey background. The only difference between the stages was the sampling frequency set, which was 10, 5 and 1 min, respectively. This was due to the apparent disturbance observed during the first stage (Figure D.1A inlet). As this disturbance was consistent across all 8 channels and mirrored in the potential measurements (Figure D.2), a hardware problem was the most likely cause. Therefore, the sampling period was reduced to 5 min in order to verify if the change was gradual or not. After it occurred once more during the second stage (Figure D.1B inlet), even though it was to a lesser degree, the sampling was further reduced to 1 min.

The current measured was in the range of 2 to  $-2 \mu\text{A}$  and changed from positive to negative



**Figure 3.2** Current over time for the different electrochemical cells. The grey area refers to the second period monitored. The periods only differ in the frequency of the data acquisition. Data disturbance was observed across all channels and a hardware problem is suspected as the source. A, abiotic (reds); B, biotic (blues). Note the difference in y-axis ranges.





**Figure 3.3** Estimation of the number of electrons transferred. The x-axis stand for  $nC_{\text{net}}$ ,  $nC_{\text{abs}}$  and  $nC_{\text{diff}}$ , respectively. Red, abiotic electrochemical cells; blue, biotic electrochemical cells.

in all cells. Given that the measurement is close to zero and that it was similar across the biotic (B) and abiotic (A) conditions, this would suggest that the current observed is due to noise, rather than a biological phenomenon.

### 3.2.1.1 Estimation of the number of electrons exchanged

In an attempt to compare the current obtained across conditions quantitatively, the number of electrons transferred was estimated. This was achieved by integrating the current over the experimental time, as described in Section 3.4.8, using the trapezoid rule ( $nC$ , Equation 3.3).  $nC$  represents the number of Coulombs, the International System (IS) unit of electric charge, equivalent to  $6.24 \times 10^{18}$  electrons. To account for the change in sign observed, the number of Coulombs was also calculated using the absolute current value ( $nC_{\text{abs}}$ , Equation 3.4), as well as the difference between these ( $nC_{\text{diff}}$ ). Figure 3.3 represents the values found, while the complete data can be found in Table D.1. The mean  $nC_{\text{abs}}$  (sd,  $n = 4$ ) of the abiotic condition (A) was 1.2584995 (0.26189567) and 1.3792684 (0.20413841) for the biotic condition (B). A two sample t-Test (see Section 3.4.14) was performed to determine if there was a significant difference between the two conditions on the number of Coulombs calculated ( $nC_{\text{abs}}$ ). The test was performed with equal variance (F-Test statistic and p-value of 1.6459 and 0.6923) and produced the t-Test statistic of -0.7274 and a p-value of 0.4944. Therefore, there is no significant difference between the two conditions (biotic and abiotic).

Furthermore, the theoretical maximum yield of Coulombs produced from full lactate consumption by Dv ( $nC_{\text{lac}}$ ) was found to be 1041.96 C (Equation 3.8, Section 3.4.8). The measurement of about 1.3 C (less than 0.001%) would suggest that very little lactate consumption with electrodes as electron sink took place.  $\text{H}_2$  exchange would still be possible in the system.

### 3.2.2 Biofilm characterisation

Electrochemistry impedance spectroscopy (EIS) was carried out to characterise the electrodes' biofilm. Figure 3.4 and 3.5 show the bode and Nyquist plots, respectively. It can be observed that there are no apparent patterns across the electrochemical cell condition (i.e. abiotic, biotic or non-connected biotic). The OCP measured before and after EIS established that the systems (cells) were stable during their measurement (Figure D.4).

A Randles cell model was used to describe the electrochemical system in terms of the solution or electrolyte resistance ( $\mathbf{R}_S$ ), the charge transfer or polarization resistance ( $\mathbf{R}_{ct}$ ), Warburg impedance (semi-infinite diffusion) ( $\mathbf{W}$ ; not to be confused with working electrode specified as WE, W1 or W2) and the dual layer capacitance ( $\mathbf{C}_{dl}$ ; see Figure 3.13). The EIS data obtained was fitted to Randles cell as an equivalent circuit. The parameters were fitted for the data obtained before and after the current measurement. These points are referred to as **T0** and **T1**, respectively. Furthermore, the difference between the parameter values was calculated by subtracting the value calculated at **T0** from the value calculated at **T1** and it is referred to as **T1-T0**. Fitted parameter values can be observed in Figure 3.6.

A one way ANOVA was carried out to determine if the parameter values were different across conditions and time points. The F test statistic and p-value outputs have been summarised in Table 3.3. It can be observed that there was no sufficient evidence to state a significant difference between any of the conditions at any of the time points or the difference thereof. This suggests that no biofilms were formed on the electrodes, which is consistent with the current values and number of transferred electrons estimated.

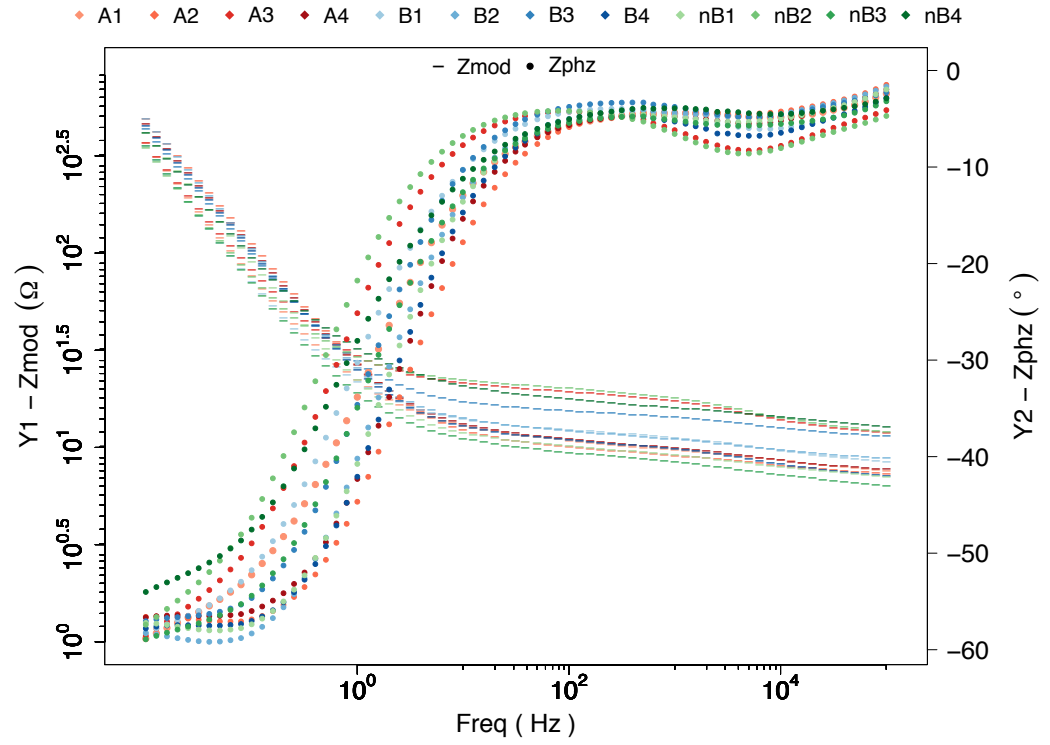
### 3.2.3 Characterisation of catalytic reactions

Cyclic voltammetry (CV) was used to identify potentials at which redox reactions were taking place. The fifth curve for each working electrode (W1 and W2) is shown in Figure 3.7. Figures D.8, D.9 and D.10 show the individual traces of the fifth curve with the estimated anodic ( $\downarrow$ ,  $E_{p,a}$ ) and cathodic ( $\uparrow$ ,  $E_{p,c}$ ) peak potentials.  $E_{p,a}$  and  $E_{p,c}$  were calculated by manually defining potential

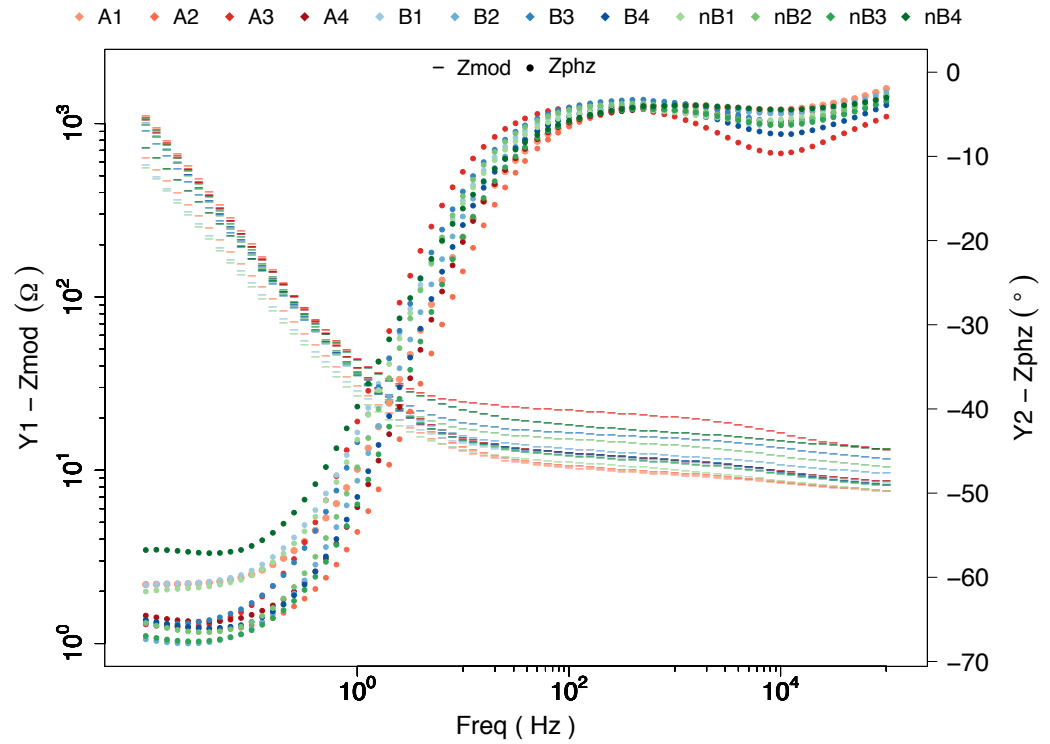
**Table 3.3** EIS model parameter comparison across conditions (A, B and nB) using ANOVA for the two time points and the difference thereof. The F test stat. test statistic and the p-values are reported for the parameters fitted on the data obtained before the current measurement (**T0**), after the current measurement (**T1**) and of the difference between them (**T1-T0**).

	$\mathbf{R}_S$		$\mathbf{R}_{ct}$		$\mathbf{W}$		$\mathbf{C}_{dl}$	
	F test stat.	p-value	F test stat.	p-value	F test stat.	p-value	F test stat.	p-value
<b>T0</b>	0.1727	0.8441	0.4076	0.6769	0.5694	0.5850	1.2940	0.3207
<b>T1</b>	0.9492	0.9492	0.9923	0.9923	0.6921	0.6921	0.1201	0.1201
<b>T1-T0</b>	0.4313	0.6624	1.0814	0.3794	0.5824	0.5783	1.0661	0.3841

A

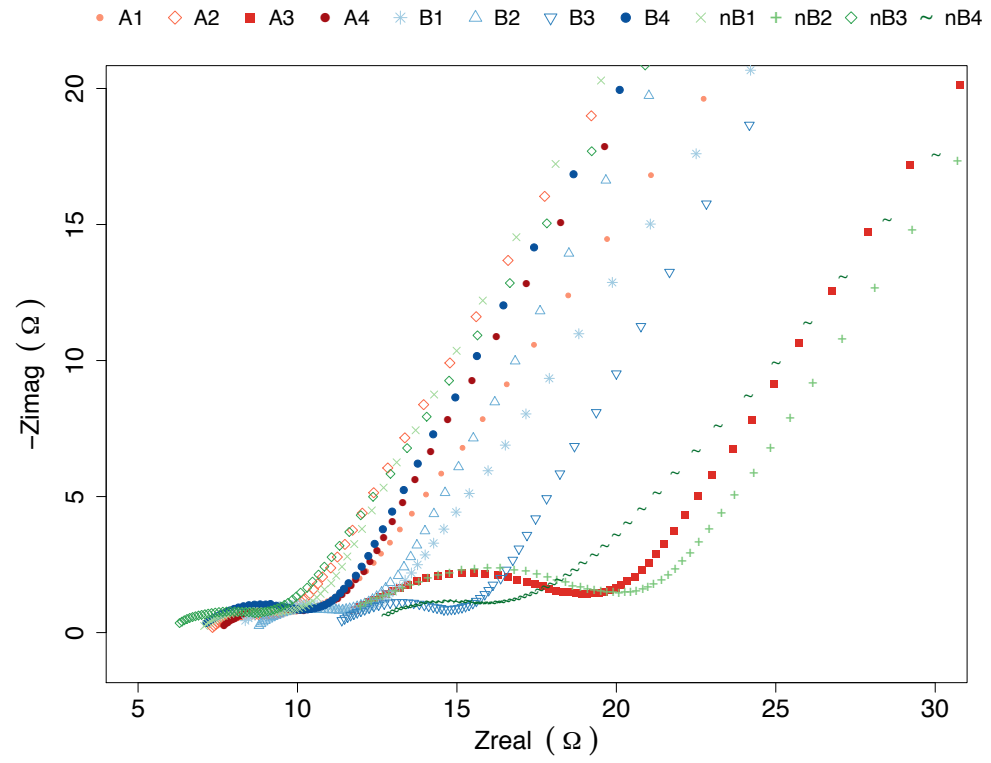


B

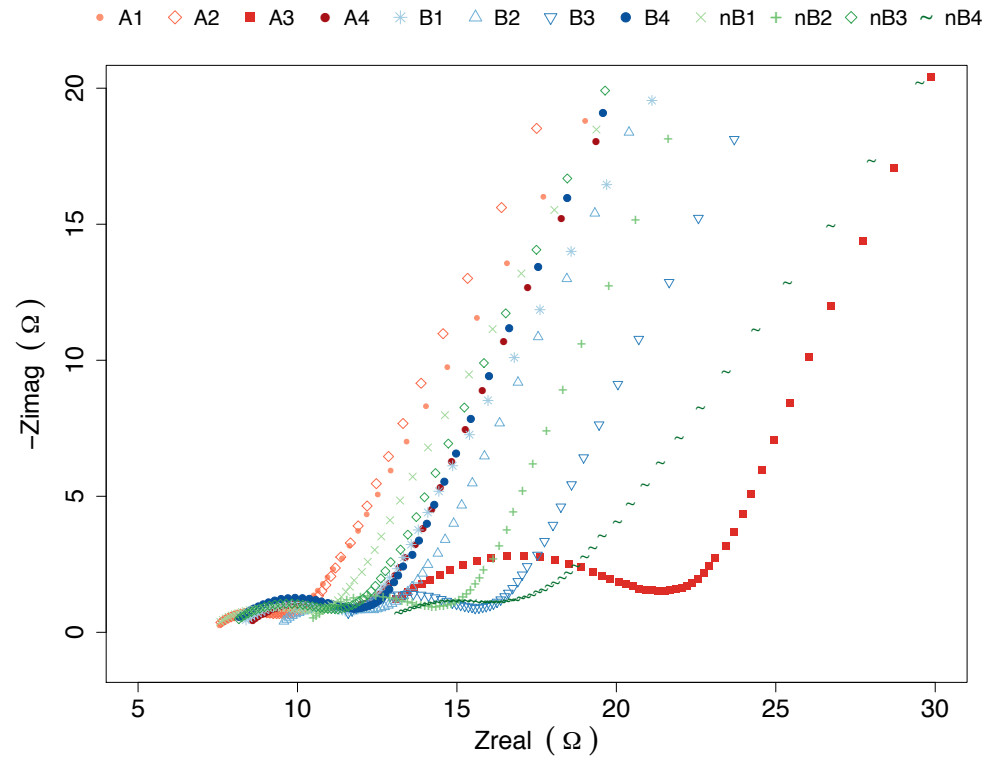


**Figure 3.4** EIS bode before (A) and after (B) current monitoring. A, abiotic (reds); B, biotic (Dv and Mm inoculated and connected; blues); nB, non-connected biotic (Dv and Mm inoculated, but not connected; greens).

A

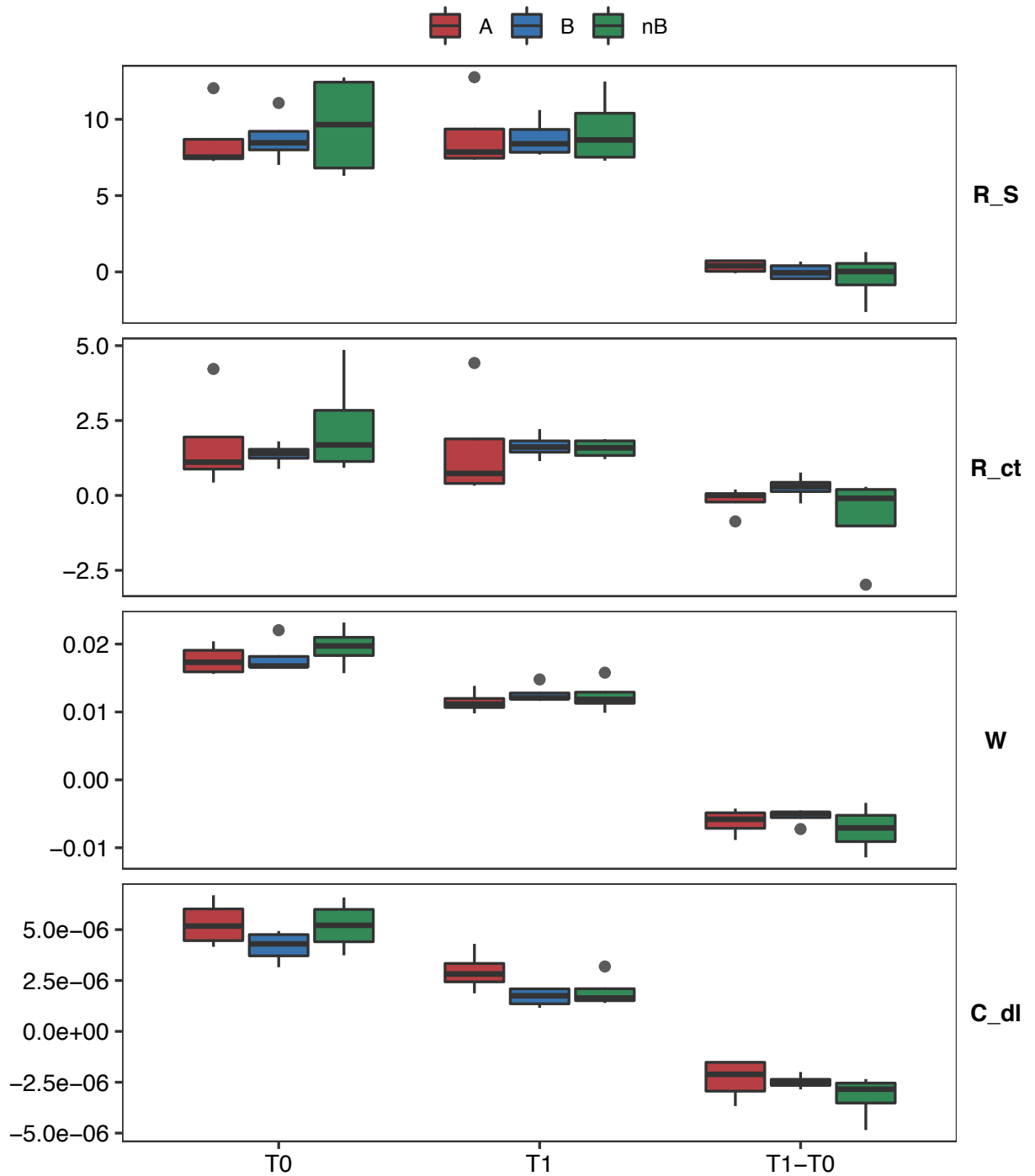


B



**Figure 3.5** EIS Nyquist before (A) and after (B) current monitoring. A, abiotic (reds); B, biotic (Dv and Mm inoculated and connected; blues); nB, non-connected biotic (Dv and Mm inoculated, but not connected; greens).

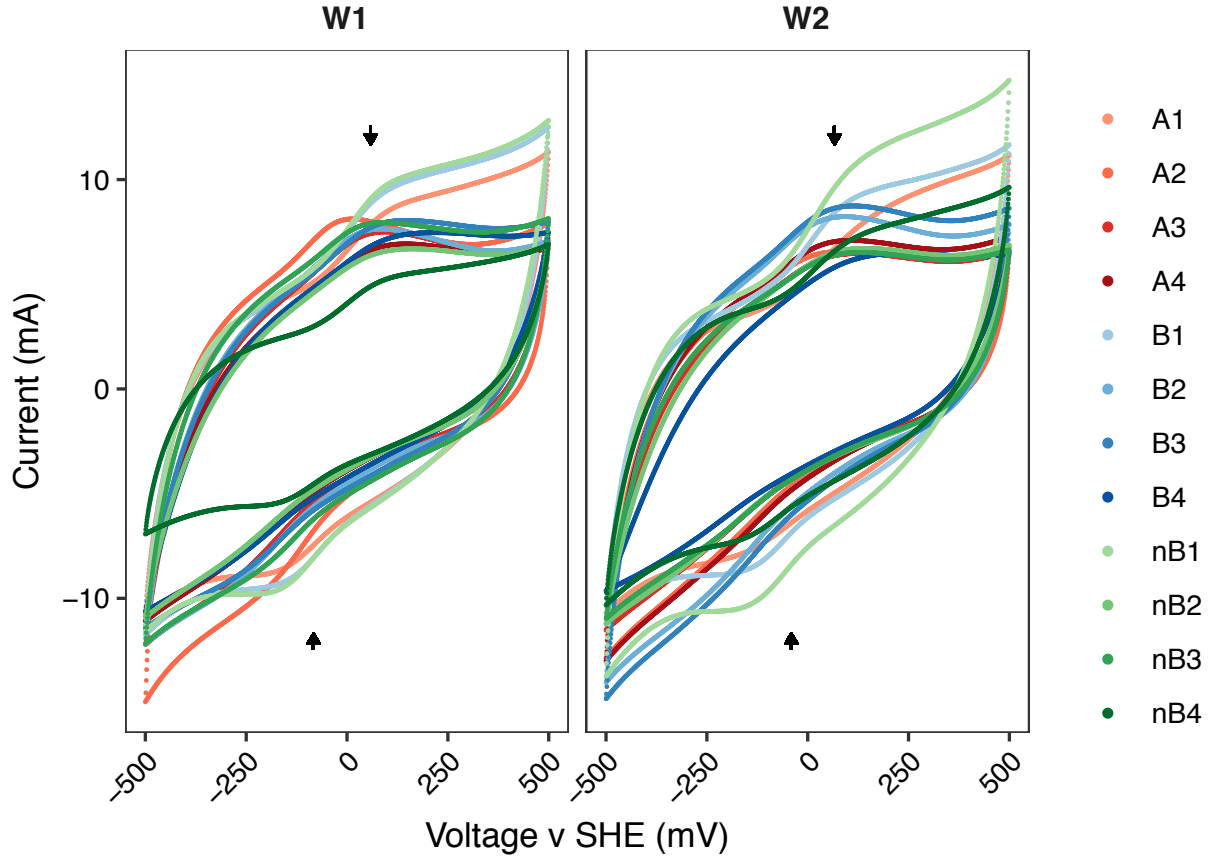
ranges and using Code 3.1 (described in Section 3.4.10). The peak potentials ( $E_{p,a}$ ) were generally well identified when present and 0 mV was calculated for missing cathodic peaks (e.g. see nB2 curve in Figure D.10). The cathodic peaks were clearly visible in most of the traces, although some



**Figure 3.6** EIS – fitted parameter values of Randles cell model shown for each time point (before and after current measurement, **T0** and **T1**, respectively) or the difference (**T1-T0**) thereof for each of the conditions. There was no sufficient evidence to state a significant difference between any of the conditions at any of the time points or the difference thereof (P values in Table 3.3).  $R_S$  is the solution resistance,  $R_{ct}$  is the charge transfer or polarization resistance,  $W$  is the Warburg impedance (semi-infinite diffusion), and  $C_{dl}$  is the dual layer capacitance. A, abiotic (red); B, biotic (blue); nB, non-connected biotic (green);  $n = 4$ .

were more pronounced. If catalytic reactions were present through biological activity (such as, but not limited to, biofilm formation), we would expect a noticeable difference to be perceivable between the biotic (B, blues; Figure D.9) and the abiotic (A, reds; Figure D.8) condition cells. This difference was not that well marked, suggesting an absence of biological activity.

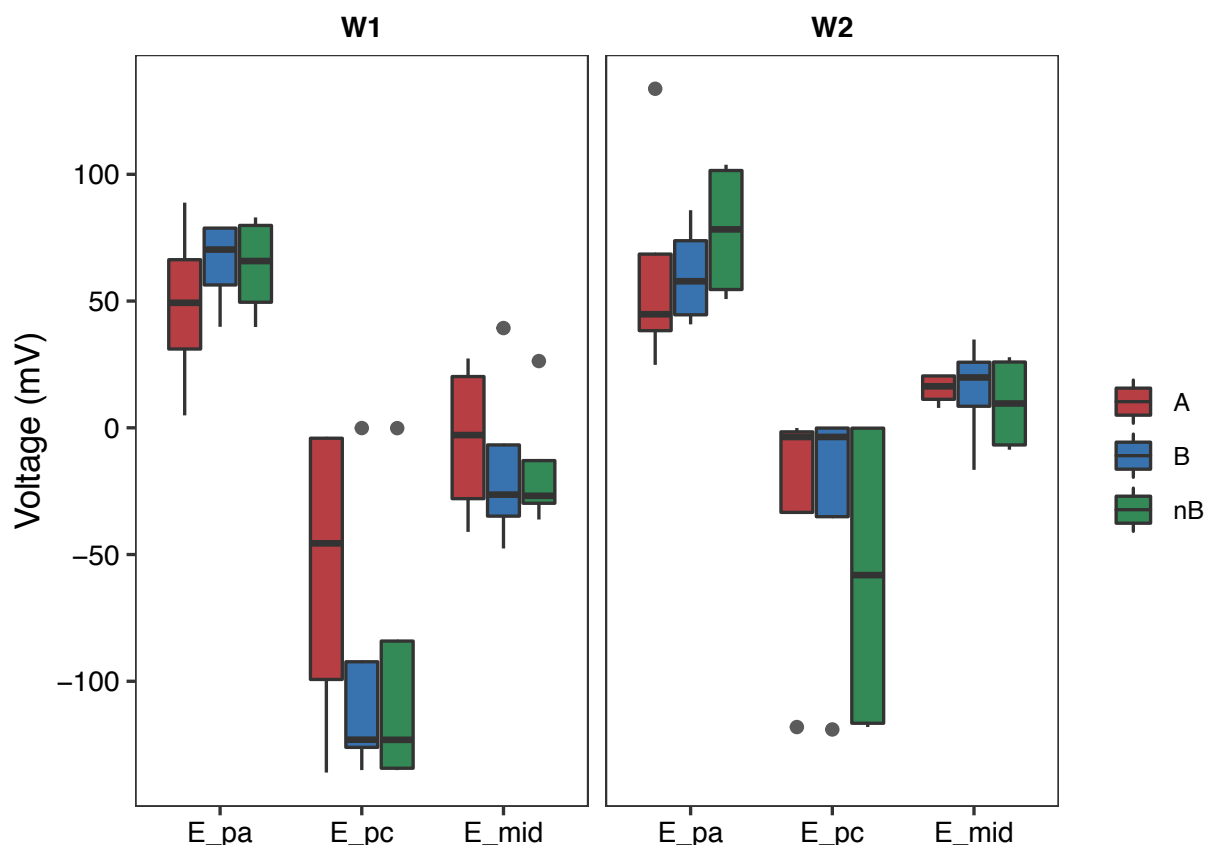
The mid-peak potential ( $E_{mid}$ ) was derived from  $E_{p,a}$  and  $E_{p,c}$ . The values for each condition and electrode can be seen in Figure 3.8. A two way ANOVA was carried out for each peak



**Figure 3.7** CV curves for each working electrode (W1 and W2) of the electrochemical cells. The fifth curve for the CVs are shown. An arrow was placed at the mean  $E_{p,a}$  and  $E_{p,c}$  for W1 (58.82712 mV and -82.83258 mV) and W2 (66.79372 mV and -40.51508 mV), which are close to the majority of the catalytic peaks of both the forward and reverse directions. W1/W2, working electrode 1/2; SHE, standard hydrogen electrode; A, abiotic (reds); B, biotic (Dv and Mm inoculated and connected; blues); nB, non-connected biotic (Dv and Mm inoculated, but not connected; greens). The numbers following the condition labels refer to the replicate number.

**Table 3.4** Estimated mean potential ( $E_p$ ) and current ( $I_p$ ) peaks and their tolerance ( $tol$ ). A, abiotic; B, biotic; nB, non-connected biotic.

CONDITION	ELECTRODE	$E_{p,a}$	$I_{p,a}$	$tol_{p,a}$	$E_{p,c}$	$I_{p,c}$	$tol_{p,c}$
A	W1	35.16	7.60	0.06	65.15	-5.88	0.63
B	W1	18.46	7.87	0.03	63.73	-6.50	0.45
nB	W1	20.75	7.20	0.03	64.35	-6.23	0.47
A	W2	48.75	6.88	0.02	57.84	-4.92	0.46
B	W2	21.07	7.80	0.08	58.38	-5.58	0.37
nB	W2	28.37	7.56	0.02	67.51	-6.12	0.53

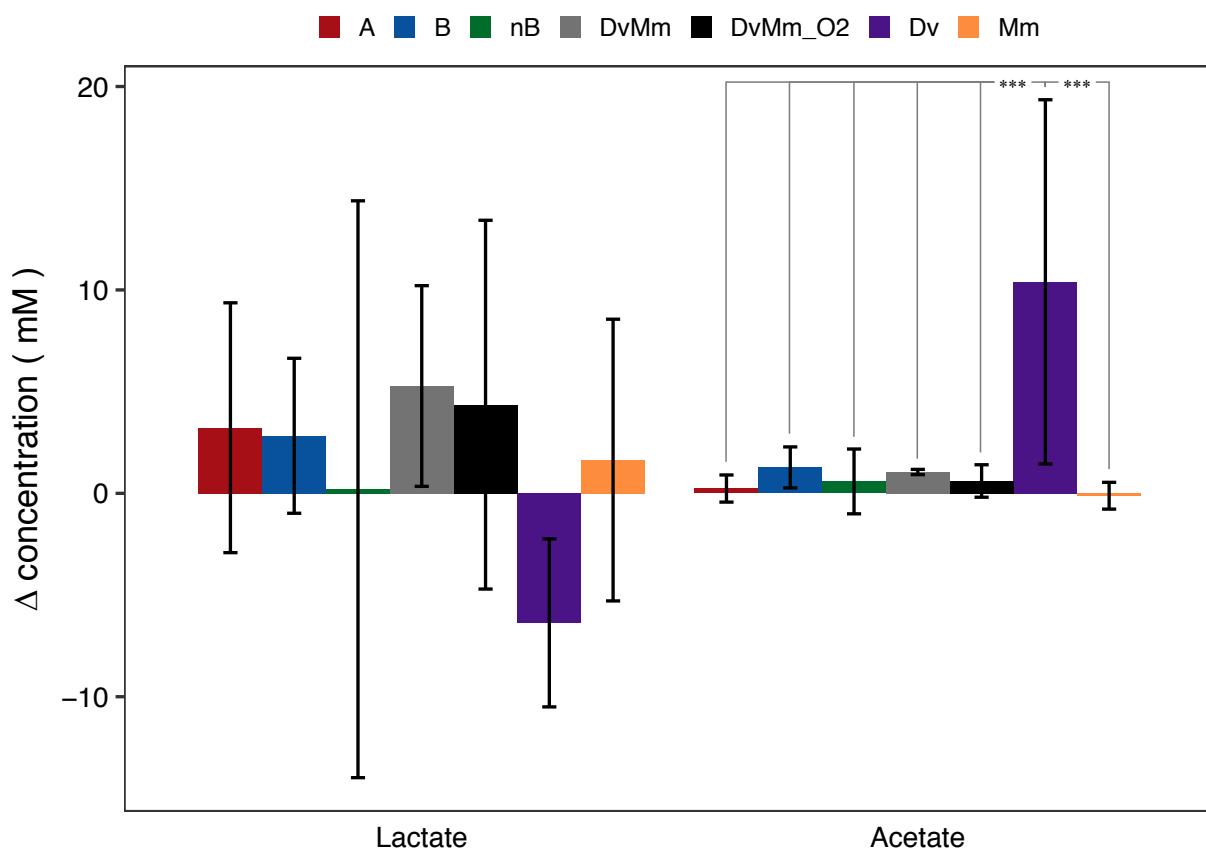


**Figure 3.8** Boxplot of the peak potentials of the CV trace of the fifth curve for each working electrode (W1 and W2) on all three conditions. The arrows indicate the anodic ( $\uparrow$ ,  $E_{p,a}$ ) and cathodic ( $\downarrow$ ,  $E_{p,c}$ ) peak potentials estimated with the `first_derivative_inflection()` function (Code 3.1).

potential ( $E_{p,a}$  and  $E_{p,c}$ ) to evaluate the following three sets of null hypotheses: (i) the mean peak potential were the same across the conditions (A, B and nB), (ii) the mean peak potential were the same across the working electrodes (W1 and W2) and (iii) there was an interaction between the conditions and the electrode. Note that the same test was not performed on  $E_{mid}$  as it was derived from the other two. The three sets of null hypothesis failed to be rejected for both  $E_{p,a}$  and  $E_{p,c}$ . The F test (p-value) values for  $E_{p,a}$  were 0.5195 (0.6035), 0.4056 (0.5322) and 0.2385 (0.7903) for hypothesis (i), (ii) and (iii), respectively. The F test (p-value) values for  $E_{p,c}$  were 0.5345 (0.5950), 2.7135 (0.1168) and 0.1870 (0.8310) in the same order. This suggests that there was no difference in the electron transfer reactions in the different electrochemical cells, regardless of whether they were inoculated or not. This is consistent with the observations made of the small current registered, the very small number of electrons transferred (compared to the theoretical maximum) and the absence of change in the EIS before and after the current was monitored.

### 3.2.4 Characterisation of Mm's metabolic activity

Based on the theoretical gas production calculation (Section 3.4.3), we expected 25.06 mL of methane to be produced. The aim of performing a gas chromatography (GC) analysis was to



**Figure 3.9** Change in compound concentration across conditions. The change in acetate concentration for the Dv control culture was significantly different from the remaining cultures. A, abiotic; B, biotic; nB, non-connected biotic; DvMm, coculture (DvMm) in sealed tubes (growth control); DvMm\_O2, coculture (DvMm) in exposed tubes (environment control); Dv, Mm, monocultures of Dv and Mm (growth controls). Error bars show the standard deviation. p-value symbols: \*, <0.05; \*\*, <0.01; \*\*\*, <0.001.

calculate the methane yield from the lactate consumed and, thus, Mm's activity. However, no detectable gas was collected in the gas bags connected to the electrochemical cells, consistent with little or no growth of the microorganisms. Therefore, the GC was not carried out.

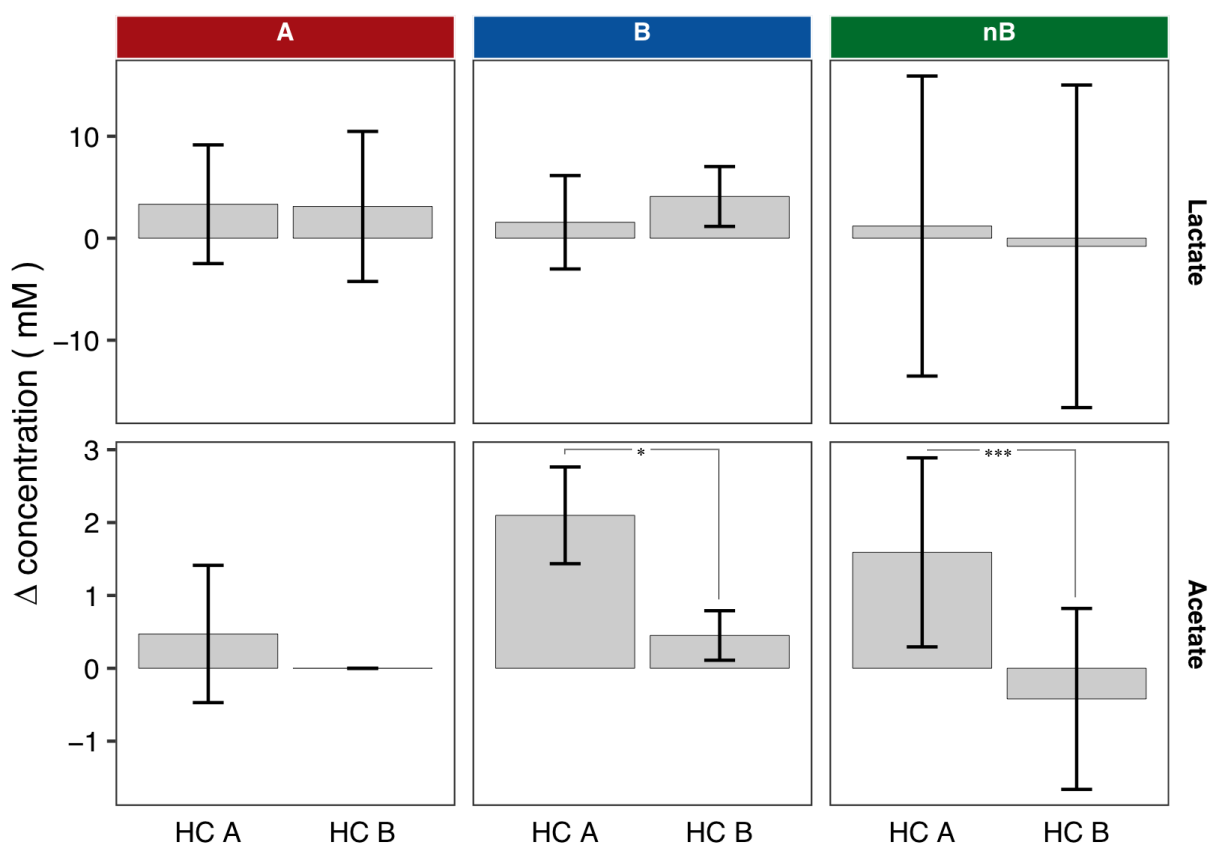
### 3.2.5 Characterisation of Dv's metabolic activity

As we were unclear about the magnitude of the current to expect, chemical analysis was performed to quantify the amount of lactate consumed and acetate produced. Therefore, the chemical concentrations could be used to verify Dv's activity. Since the cultures were inoculated with late-phase cultures, there was a chemical carry-over that was culture dependent. To account for this, samples were taken at the start of the experiment (after inoculation;  $T_0$ ) and at the end ( $T_{end}$ ). Ion chromatography was carried out to quantify the lactate and acetate concentrations in the samples (see Section 3.4.12). The change ( $\Delta$ ) in concentration ( $c$ ) was calculated by subtracting the concentration at the start from that at the end as described in Equation 3.16. It was to be expected that the change in lactate concentration ( $\Delta lac$ ) would be negative ( $-$ ), as it is consumed by Dv, while



the change in acetate ( $\Delta ace$ ) was expected to be positive (+) as it is one of Dv's by-products. The only exceptions to this would be for Mm control cultures and abiotic electrochemical cells, where I expected the change of both compounds to be zero. However, it was unclear if this was to be true for the half-cells containing Mm (half-cell A; HCA), as the diffusion could be slow due to the presence of the cellulose membrane between the half-cells and absence of shaking.

Figure 3.9 shows the change in compound concentrations for the different conditions. A negative change in lactate accompanied by a positive change in acetate ( $-\Delta lac / +\Delta ace$ ) was



**Figure 3.10** Change in compound concentration data by treatment type. The change in lactate and acetate for each half-cell (HC) is shown ( $n = 4$ ). Paired t-Tests were carried out to compare the change in compound across half-cells. Both the biotic (B) and non-connected biotic (nB) had a significant difference in the change of acetate concentration, as indicated by the p-value symbols. The abiotic (A) condition showed no significant difference between half-cells for either compound. p-value symbols: \*,  $<0.05$ ; \*\*,  $<0.01$ ; \*\*\*,  $<0.001$ .

**Table 3.5** Post hoc output of significant results for change in acetate concentration across conditions. CI, confidence interval; Adj., adjusted; \*\*\*, p-value $<0.001$

Comparison	Estimate	CI (low)	CI (high)	Adj. p-value	symbol
Dv - A	10.162356	5.176355	15.148357	0.000005	***
Dv - B	9.123164	4.137162	14.109165	0.000035	***
Dv - DvMm	-9.349861	-15.363205	-3.336517	0.000451	***
Dv - DvMm_O2	-9.792059	-14.701933	-4.882184	0.000007	***
Dv - Mm	-10.516234	-16.529577	-4.502890	0.000075	***
Dv - nB	-9.812598	-14.798600	-4.826597	0.000010	***

observed for the Dv control cultures. A one way ANOVA was performed to determine if the change in compound concentration was the same for all conditions or not. Only  $\Delta_{ace}$  for the Dv monoculture was significantly different from the rest, as indicated in Figure 3.9, with an F-Test statistic of 8.166022 and a p-value of 0.000015. Table 3.5 contains the statistical output for those conditions that were different as determined by the post hoc test (see Methods Section 3.4.14). The complete statistical outputs are included in Section D.4.3. This indicates that the Dv cultures were the only ones growing as they were the only ones with evidence of metabolic activity.

As mentioned before, it was unclear whether a change in compound concentrations could be expected in both half-cells or only in half-cell B, where Dv was inoculated, due to limited diffusion. To evaluate this, the change in chemical concentrations across half-cells of the same condition was compared by using a paired two sample t-Test (see Methods Section 3.4.14). A significant difference was identified across the half-cells of the biotic (B) and non-connected biotic (nB) cells with the t-Test statistic (p-values) of 5.7721 (0.01034) and 7.6276 (0.004678), respectively. Surprisingly, however, a higher acetate concentration was found in half-cell A, which housed Mm. An accumulation of acetate to be found in either both half-cells or more evidently in half-cell B had been expected, since Dv would be producing it and diffusion to half-cell A would be slow. Moreover, lactate seemed to have increased in most conditions, particularly in the abiotic cells. Taken together, this would suggest that there has been some evaporation as neither Dv nor Mm are lactate producers. To determine whether the evaporation occurred during the experiment or during the IC sample preparation and measurement, technical replicates should be included in future analyses. Additionally, the chloride peak could be used to correct for evaporation events.

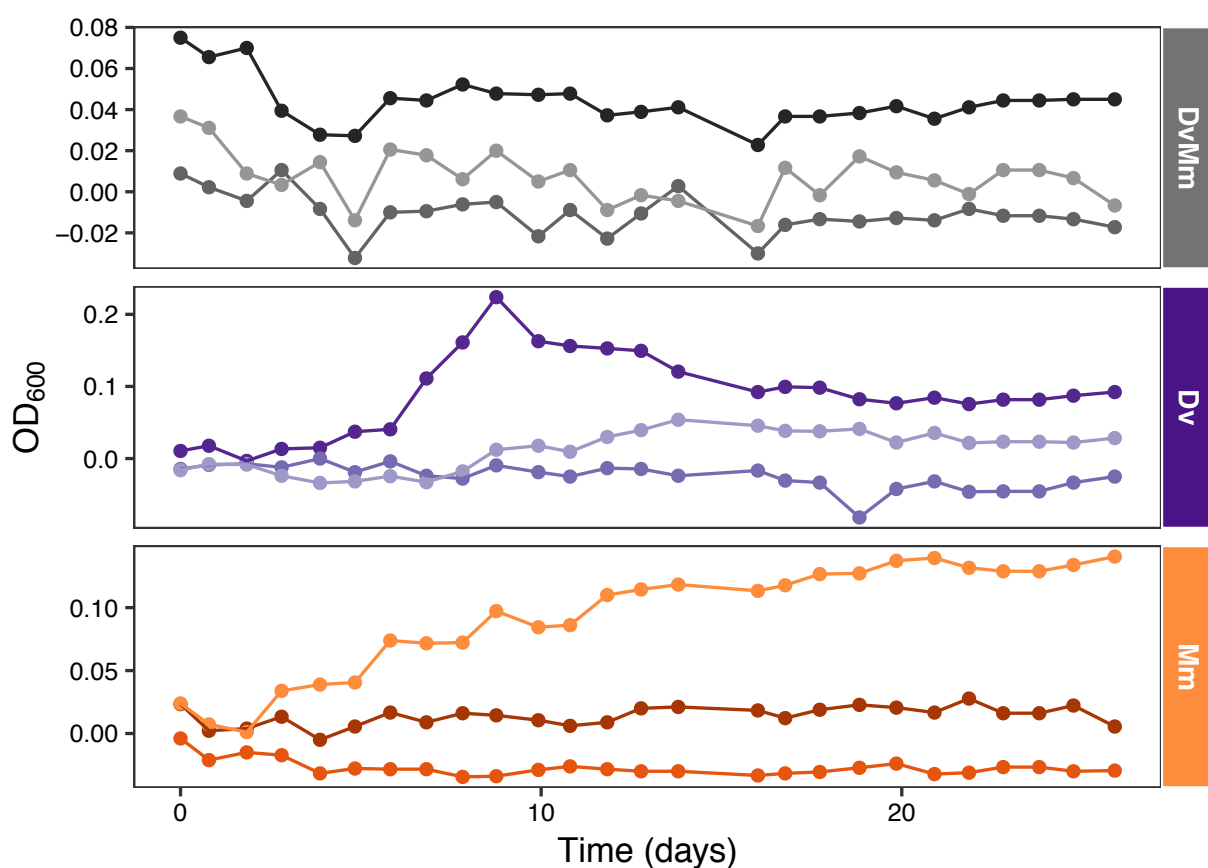
### 3.2.6 Bacterial growth

Dv and Mm are normally grown at 37 °C (Großkopf et al., 2016; Walker et al., 2009), but there were technical limitations in implementing temperature control in the experimental electrochemical platform used for this work (described in Chapter 2). Therefore, control culture tubes were included within the experimental design to monitor the growth achieved at room temperature. The optical density at 600 nm ( $OD_{600}$ ) of the control culture tubes was measured daily. Note that the seed cultures were successfully grown at 37 °C in the same medium, as evidence by an increase in turbidity and the formation of biofilm visible in the Mm monocultures.

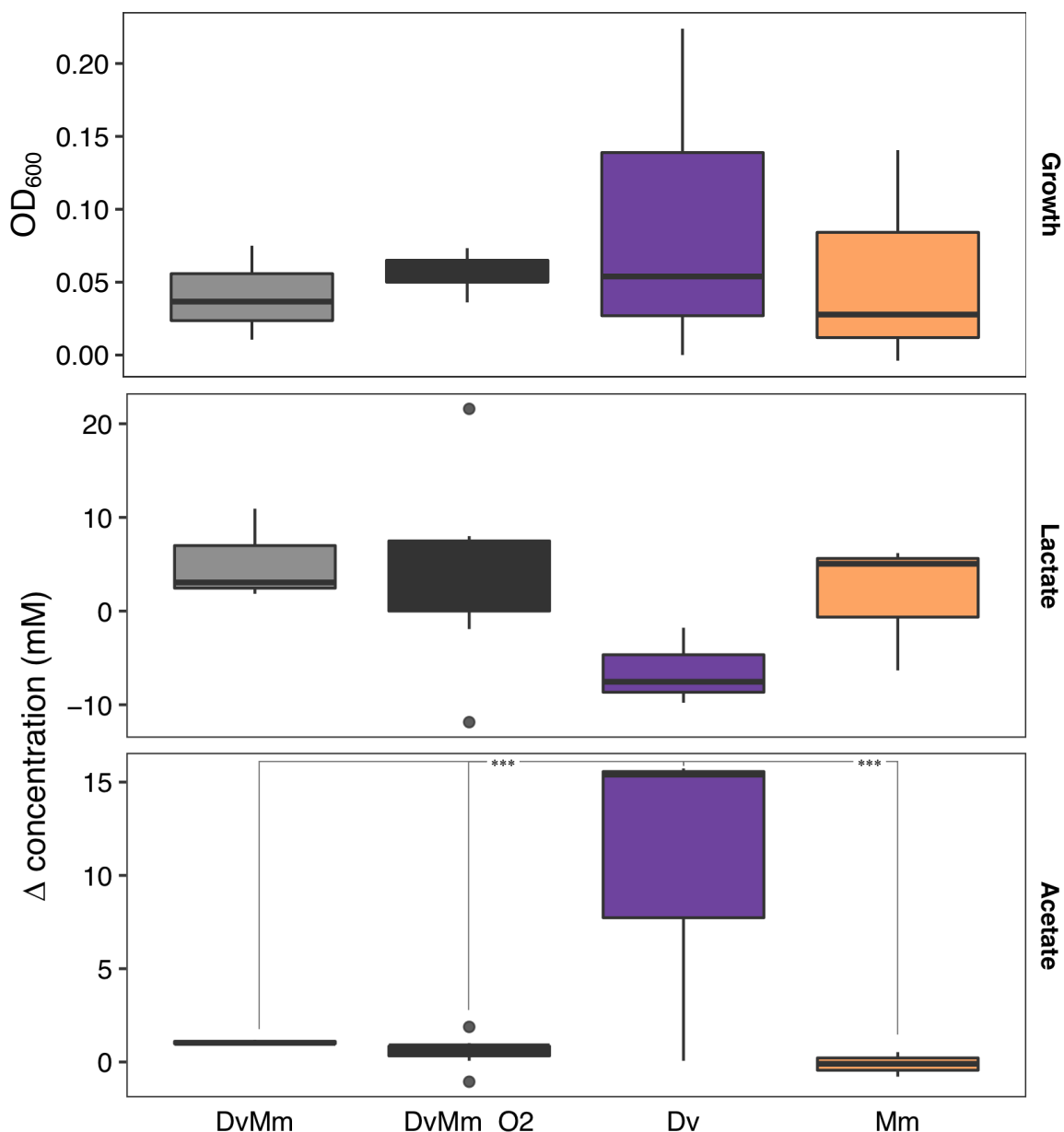
Figure 3.11 shows the growth of the monitored controlled tubes for Dv, Mm and DvMm. It can be observed that the cultures did not thrive at room temperature, except for one of the Dv replicate cultures, which achieved a maximum  $OD_{600}$  of ca. 0.2 after approximately 9 days. Figure D.14 shows unpublished data obtained by Dr. Jing Chen of DvMm, Dv and Mm cultures grown at 37 °C and subcultured twice, resulting in 3 periods. These data show that Dv and

DvMm reached an OD<sub>600</sub> of ca. 0.2 in the first week of culturing, similarly to the Dv monoculture that grew as seen in Figure 3.11. One of the replicates of the Mm monocultures seemed to be slowly growing, although the maximum OD<sub>600</sub> achieved when grown at 37 °C was over 0.5. These results strongly suggest that the microorganisms were not able to grow at room temperature in the time period studied.

The OD<sub>600</sub> values obtained were further compared with the change in compound concentration measured with ion chromatography (see Result Section 3.2.5 and Materials and Methods Section 3.4.12). The highest OD<sub>600</sub> for each condition replicate for the DvMm, Dv and Mm cultures, as well as the end point OD<sub>600</sub> measurement of the DvMm\_O2 control tubes ( $n = 9$ ) are shown in the top panel of Figure 3.12. The second and third panels contain the change of lactate and acetate concentrations, respectively, observed. The change in acetate concentration was significantly higher for the Dv monoculture compared to all other conditions, as presented in Result Section 3.2.5. This further supports that only the Dv condition achieved some growth.



**Figure 3.11** Growth curves of control cultures at room temperature (RT; ca. 21 °C). The three replicate for each condition (DvMm, Dv and Mm) are shown. Figure D.13 shows the mean OD<sub>600</sub> with error bars. However, the low OD<sub>600</sub> achieved means that a high standard deviation was measured and hence the graph is not a clear representation of the growth achieved.



**Figure 3.12** Comparison of control tube growth with change in lactate and acetate concentration. The top panel shows box plots for the maximum  $OD_{600}$  value recorded for the DvMm, Dv and Mm cultures ( $n = 3$ ) and the final  $OD_{600}$  value for the DvMm\_O2 tubes ( $n = 9$ ). The second and third panel show the change in lactate and acetate concentration calculated, respectively, for each of these conditions. The change in acetate concentration was found to be significantly different between Dv and all other conditions as described in Section 3.2.5. p-value symbol: \*\*\*,  $<0.001$ .

### 3.3 Discussion

The aim of this chapter was to investigate the “syntrophy over wires” hypothesis presented in this work. This hypothesis proposes that Dv and Mm can grow syntrophically on separate electrodes due to electron transfer between them in lieu of molecular hydrogen exchange. To the author’s knowledge, this type of interaction has not been previously described in the literature. Therefore, a range of techniques were combined to gain as much insight into this electrochemical system as possible.

The current between the working electrodes (WEs) was monitored to record the electron transfer between Dv and Mm. Low current values (ca.  $-1\ \mu\text{A}$  during the last week of experimental data) were obtained and, hence, the estimated number of Coulombs was only about 0.001% of the theoretical maximum calculated for complete lactate consumption (Figures 3.2 and 3.3). Miller et al. (2016) reported a current recorded for *Shewanella oneidensis* that reached  $15\ \mu\text{A}$ . It was expected that the abiotic cells to record current values close to zero, while the biotic ones to produce a positive current that would increase proportionally to the microorganisms' growth, resulting in a trace similar to a typical bacterial growth curve as observed in (e.g. Marsili et al., 2008; Miller et al., 2016). However, there was no difference between the current traces observed for the abiotic or biotic conditions, as these were effectively zero across conditions. Taken together, the absence of current suggests that no growth occurred.

Some data disturbance was observed in the recording of the current during the first two monitoring stages (Figures 3.2 and D.1). Two likely explanations are either a hardware problem or the occurrence of chemical reactions on the electrode(s) that could be investigated in future work (Lefrou et al., 2012; Marsili et al., 2008; Scholz, 2010). To determine whether it was caused by the former, the system could be set up with eight identical resistors that would mimic the electrochemical cells and the current monitored as described here (priv. comm. with potentiostat manufacturer).

There would be two likely sources for the disturbance caused by chemical reactions taking place on the electrodes: either inorganic reactions or reactions catalysed by the microorganisms. The latter is unlikely since the time-scale was small and the changes took place in all the cells, regardless of the presence of microorganisms. Therefore, only inorganic reactions are likely to have been responsible as an alternative to the hardware problem. The inorganic reactions could be the corrosion of media components, such as the resazurin or cysteine present in the medium (Marsili et al., 2008). These components are commonly excluded from the medium in electrochemical systems (e.g. Lohner et al., 2014; Marsili et al., 2008; McAnulty et al., 2017; Milliken and May, 2007), although there have been studies that preserved their use (e.g. Nichols et al., 2015). However, vitamins and trace metals are essential for both microorganisms' growth, as well as strict anaerobic conditions only achievable by the use of chemical reductants (i.e. cysteine). Future experiments could dispense the use of resazurin and, if needed, implement the use of an oxygen monitoring probe instead, such as the autoclavable polarographic oxygen electrode (Mettler Toledo, Columbus, OH) (Hou et al., 2010) or a Microx TX3 trace instrument (PreSens, Munich, Germany) (Okamoto et al., 2015).

Electrochemical impedance spectroscopy (EIS) was carried out before and after the current between the working electrodes was monitored and Bode and Nyquist plots (Figures 3.4 and 3.5)

were presented. A Randles cell model was used as an equivalent circuit to model the electrochemical system present in the cells. The fitted parameter values were compared across time points and conditions (Figure 3.6). However, no significant difference was detected, consistent with the lack of growth suggested by the low current values. In the future, focus should be given to the task of ensuring that the working electrodes are consistently placed at the same distance to ensure that the electrical resistance of the electrolyte is constant across electrochemical cells (Dewan et al., 2008). 3D printed structures to allow consistent placement of the electrodes within the cells would be a feasible, low-cost, effective technique (Bauer and Kulinsky, 2018; Chae et al., 2015; Rusling, 2018).

EIS allows some insight into the resistance of the electrochemical system to be gained. The electrochemical system presented here was viewed to extend from the W1 to the W2, unlike conventional single bioelectrode (either bioanode or biocathode) systems (e.g. Dewan et al., 2008; Marsili et al., 2008). As such, the EIS measurements were performed across the working electrodes, connecting W1 to the Gamry W/WS leads and W2 to Gamry's C/CS leads. In the results presented previously, it was assumed that the Randles cell model was able to explain the resistance across the working electrodes. Future work would be well spent in determining whether that assumption is valid and if there are any differences in the insights to be gained.

Cyclic voltammetry (CV) was used to characterise the redox reactions occurring at a range of potentials in the electrochemical cells. An algorithm was developed to automate the identification of the anodic and cathodic peak potentials in order to estimate the mid-peak potential of the systems. This successfully identified clear peaks as can be observed in Figures D.8, D.9 and D.10. Even though the peaks could have been manually identified using the Gamry Analysis Software, implementing an algorithm within the R environment to do so facilitates the downstream analysis of the data obtained. This is particularly important when the data set size increases due to the larger experiment designs, achievable in the electrochemical platform used for this work and described in Chapter 2. The mean peak potentials for the two WEs and the different conditions were summarised in Table 3.4. Once more, no significant difference was observed between conditions, specially between abiotic and biotic cells, which suggests that the peaks observed are due to inorganic chemical reactions rather than biologically catalysed ones.

Mm's metabolic activity could not be evaluated since no gas accumulation was evident in the gas collection bags. This in itself suggests that there was little or no methane produced, which indicates that Mm did not grow. The absence of gas accumulation in the gas collection bag highlights a limitation of the electrochemical platform used, described in Chapter 2. Alternatives to gas collection and analysis could be considered, such as the implementation of a water displacement column, similar to the one described by Oscar et al. (1987), or the connection to a GC autosampler.

Ion chromatography (IC) was used to characterise Dv’s metabolic activity, by quantifying the lactate and acetate concentrations. The change in compound concentration was calculated by subtracting the concentration found after inoculation from that found at the end of the experiment. A negative change in lactate accompanied by a positive change in acetate ( $-\Delta lac / +\Delta ace$ ) was only observed for the Dv control cultures, suggesting that the Dv cultures were the only ones to grow as they were the only ones with evidence of metabolic activity.

The comparison across the electrochemical half-cells did not show any metabolic activity. A  $+\Delta ace$  proved to be significantly different across half-cells for the biotic and non-connected biotic set of electrochemical cells. Surprisingly, however, the higher  $+\Delta ace$  was found in half-cell A (HC A), even though this would have been expected to happen in HC B, where Dv was inoculated. Moreover, a  $+\Delta lac$  was observed across all conditions in some of the replicates. It is, therefore, unclear if this compound concentration changes are, in effect, a result of metabolic activity or if it could be due to an increase in compound concentration in the electrochemical cells through water-loss or osmosis or due to evaporation of the samples obtained or of the IC samples prepared for analysis. It is unlikely that the change of compound concentration is due to the IC methodology itself as it is not evident in the IC data for the control Dv cultures. Including technical replicates and implementing a chloride peak normalisation could help dispute this point in future experiments.

Taken together, the results presented in this chapter suggest that no growth was achieved by the microorganisms, except for one of Dv’s control cultures as supported by the IC data and the growth curves of control cultures. The lack of nutrient homogeneity resulting from the nature of the batch mode used when culturing the microorganisms could have affected the microorganism’s ability to grow. The operating mode called ‘batch’ refers to the electrochemical cells being filled with fresh medium and inoculated at the start of the experiment without the addition of medium throughout. This type of operating mode, although often chosen for its practicality, results in lower yields due to nutrient consumption and it can be thought of as a decaying system (Doelle et al., 2009). Although the Dv, Mm and DvMm cultures have been grown in Hungate tubes (batch mode), the internal volume of the electrochemical cells, which is ca. 20 times larger than the Hungate tubes, could cause larger nutrient gradients, creating nutrient-deficient zones around the microorganism’s biomass. This is a common result from a lack of stirring characteristic of cultures grown in batch mode. However, the generation of nutrient-deficient zones would be coupled with a decrease in the lactate concentration measured by the IC, which was not observed. Therefore, the nature of the operation mode was probably not the cause for the absence of growth. Nevertheless, if growth had been achieved, it would have been sub-optimal. The more common solutions which include the addition of stirring mechanisms, such as mechanical stirring or air lift

(pneumatic stirring) (Meyer et al., 2016). Yet, these would be practical difficult to implement in the electrochemical platform. In order to tackle this, the mode of operation could be changed to a continuous feed system, where fresh medium is continuously pumped and metabolised culture broth is removed at the same rate, maintaining a constant and homogeneous volume (Doelle et al., 2009). The electrochemical cell design could be easily modified to enable a continuous operation mode by inserting ports at the side and top of each half-cell in the electrochemical cells. The fresh medium would be pumped at the bottom and the metabolised broth would be removed at the top.

An alternative reason for the absence of microbial growth would be their inability to attach to and interact with the electrodes. As mentioned previously, Mm has been used in bioelectrochemical experiments (Lohner et al., 2014). Organisms of the same phylogeny as Dv have been found in BES inoculated with environmental samples (*Desulfovibrio* sp. A2 and *Desulfovibrionales* Pisciotta et al., 2012; Sharma et al., 2013, respectively) and Dv was identified to the species level in the work performed by Croese et al. (2011). However, Croese et al. (2011) cultured Dv at the cathode, posing the question of whether Dv would be able to grow on a bioanode. The author acknowledges that control experiments should have been carried out prior to the investigation of the “syntrophy over wires” hypothesis in order to establish whether Dv and Mm are capable of interacting with the electrodes in the way intended. Such a control experiment would involve culturing each organism as a monoculture in one of the half-cells in the electrochemical cell. The two working electrodes in the cell would be connected through the potentiostat. Rather than measuring the current between the working electrodes, a potential would have to be set across the working electrodes. Dv would have to be grown on the anode, with a potential that would allow Dv to use the electrode as a TEA, while Mm would have to be grown on the cathode, with a potential that would enable Mm to use the electrode as an electron source. The ‘ideal’ potential for the organism’s grow would be determined by having multiple cultures with a different potential. Chromatography data would inform about the metabolic activity observed, while CV and EIS would be used to characterised the reactions and biofilms observed. Although these control experiments should be performed, it would be reasonable to assume that the electrode attachment and interaction did not significantly affect the organism’s grow during this work given the literature findings discussed previously (Croese et al., 2011; Lohner et al., 2014).

The the key factor that hampered the success of the experiment was the temperature at which the experiments were carried out. Normally, both Mm and Dv are cultured at 37 °C , and therefore, incubation at room temperature (ca. 21 °C) implies a ca. 16 °C difference. Walker et al. (2009) demonstrated that Dv uses two different pathways when grown syntrophically (i.e. H<sub>2</sub> production) versus when respiring sulphate. Syntrophic growth relies on periplasmic enzymes and temperature is known to affect the membrane and periplasm fluidity (Alberts et al., 2002).



Therefore, the reduced temperature could be restricting the membrane-bound and periplasmic enzymes essential for energy transduction in the absence of sulphate (Anthony, 1988). *Desulfovibrio desulfuricans* is a well known MFC species, which utilises  $\text{S}^{2-}$  and  $\text{SO}_4$  to interact with the electrodes, and is able to grow at room temperature (Habermann and Pommer, 1991), supporting the previous statement.

Moreover, the temperature would also have an effect in the microorganism's metabolic rates. A biochemical reaction rate constant ( $k$ ) is defined by the Arrhenius equation  $k = A e^{\frac{-E_A}{RT}}$ , where  $A$  is the frequency factor (related to the frequency of collisions between molecules),  $E_A$  is the activation energy,  $R$  is the gas constant and  $T$  is the temperature. A ten degree temperature increase results in a doubled to tripled rate constant (Upadhyay, 2006). On the other hand, reducing the temperature would significantly reduce the rate constant, reflecting that the metabolic rates of the microorganisms would be slower when grow at 37 °C. This would suggest that a longer period of time than previously anticipated (Chapter 2) is required when culturing the microorganisms at room temperature in order to achieve the same biomass yield as when grown at 37 °C. A control experiment should be performed in future work to determine the period that should be observed. Note, however, than an increase in the experimental time would be limited by the electrochemical platform's ability to maintain strict anaerobic conditions. As mentioned in Chapter 2, this was observed to be approximately 24 days. This period could be theoretically prolonged by introducing additional chemical anaerobic atmosphere generation sachets, but there would be a practical limit to the number of sachets that could be introduced.

An alternative would be to implement temperature control in the electrochemical platform. There are several solutions that could be used to achieve this. Given access to a warm room, the entire set-up could be placed within it. In the absence of such a room, a heating element could be used. As these use AC power, they produce electromagnetic noise. Therefore, it would have to be placed outside the Faraday cage. A drum heater or silicon heating pads could be implemented with the placing of a DC temperature sensor within the Faraday cage to monitor and regulate the temperature. An alternative would be to use a circulating water bath system with tubing around the containers set to the desired temperature.

Another limitation in this investigation was the lack of a positive experimental control, owing to the fact that there has not been a description of a similar electrochemical set-up described in the literature to the author's knowledge. This want of a control in light of a system that did not perform well due to temperature constraints has prevented some technical aspects being "ironed-out", such as the CV scan rate and potential range of the EIS lower frequency bands. Once the growth of the microorganisms is achieved and if our hypothesis proves to be correct, this system could be used as a positive control to find other cases of "syntrophies over wires", in addition

to the tool described in the next chapter. Furthermore, the analyses presented in this work could be extended; EIS could be employed to determine the redox reaction rates or surface area characteristics of the electrodes (Manohar et al., 2008), while CV could be used to determine the redox state and the number of transferred electrons (Scholz, 2010).

## 3.4 Materials and Methods

### 3.4.1 Experimental design

The experimental design is summarised in Table 3.6. The main experimental set-up for the purpose of this work was to have an electrochemical cell (see Figures 2.2 and 2.8 in Chapter 2) with Dv inoculated on one half-cell containing a working electrode (W1) separated from Mm, inoculated in the other half-cell containing working electrode 2 (W2), by means of a membrane. As mentioned before, the growth of both organisms was expected by an exchange of electrons through connecting the two working electrodes (WEs, i.e. W1 and W2), evaluated by monitoring the current between them in zero resistance ammeter (ZRA) mode. This is referred as the biotic (B) condition. Two additional set-ups were included as controls. The first was to determine whether the growth of the organisms was due to the electron exchange through the wire. Therefore, the same set-up as above was implemented, except that the two WEs were not connected and the current was thus not monitored and it is referred as the non-connected (open-circuit) biotic (nB) condition. An additional control was to determine if any current observed was generated by the microorganisms. Thus, a set-up without microorganisms where the WEs were connected and monitored was also included and is referred to as the abiotic (A) condition. Three Hungate tube cultures of the

**Table 3.6** Experimental design summary

	MAIN			CONTROLS								
VARIABLES	B			nB <sup>1</sup>			A <sup>2</sup>			DvMm.O2 <sup>3</sup>		
Dv	+			+			-			+		
Mm	+			+			-			+		
Connection <sup>4</sup>	+			-			+			NA		
Expected growth Dv	+			-			-			+		
Expected growth Mm	+			-			-			+		
Total no. of replicates	4			4			4			9		
Replicates per container	1	1	2	2	1	1	1	2	1	3	3	3
Container ID	I	II	III	I	II	III	I	II	III	I	II	III
System	Electrochemical cell									Hungate tubes		

<sup>1</sup> Control to establish if syntrophy ( $e^-$  transfer) occurs via wire

<sup>2</sup> Control to establish if the current observed is due to microbial processes

<sup>3</sup> Control to discard O<sub>2</sub> contamination; tubes exposed to container environment by replacing caps with permeable membrane

<sup>4</sup> Connection to potentiostat and current monitoring (ZRA mode)

**Table 3.7** Control culture tubes summary

	CONTROL CULTURE TUBES		
	Coculture	Monoculture	
VARIABLES	DvMm	Dv	Mm
Dv	+	+	-
Mm	+	-	+
Additives <sup>1</sup>	-	+	+
Number of replicates	3	3	3
System	Hungate tubes		

<sup>1</sup>Sulphate for Dv; H<sub>2</sub>, pyruvate and NaCl for Mm.

“traditional” DvMm coculture were included in each container as controls. Their cap was replaced by a permeable membrane within the anaerobic chamber to ensure the cultures were exposed to the same environment as the electrochemical cells and to test for possibility of lack of growth due to the presence of oxygen. Note that this is a stringent test, as these tubes were placed within the experiment container, while actual experimental cultures were within the electrochemical cells that provided additional isolation from oxygen. These are referred to as the DvMm\_O2 condition.

Control cultures were prepared to control for the growth of the microorganisms at room temperature (Table 3.7). These included Dv and Mm monocultures ( $n = 3$  each) and coculture (DvMm) grown in sealed tubes ( $n = 3$ ).

### 3.4.2 Microorganisms and culturing

The microorganisms used were *Methanococcus maripaludis* S2 (Mm; DSM2067) and *Desulfovibrio vulgaris* strain Hildenborough sp. s1 (Dv; DSM644, strain isolated in Großkopf et al. (2016)). Both strains were originally obtained from DSMZ (www.dsmz.de). Dv uses lactate as a carbon source, while Mm uses CO<sub>2</sub>. Dv and Mm were grown together in co-culture medium (CCM) (described in Großkopf et al. (2016), originally modified from Walker et al. (2009)). CCM consists of a basal salt solution (K<sub>2</sub>HPO<sub>4</sub>, 0.19 g/L; NaCl, 2.17 g/L; MgCl<sub>2</sub> x 6 H<sub>2</sub>O, 5.5 g/L; CaCl<sub>2</sub> x 2 H<sub>2</sub>O, 0.14 g/L; NH<sub>4</sub>Cl, 0.5 g/L; KCl, 0.335 g/L; NaHCO<sub>3</sub>, 2.5 g/L; Na-Lactate, 3.36 g/L) to which a 1000 X (1 g / L) resazurin stock solution, a 100 X trace metal stock solution (metal mix, Section C.1.1, Table C.1), a 1000 X vitamin stock solution (vitamin mix, Section C.1.2, Table C.2), and a 100 X (35.03 g / L) cysteine stock solution (Section C.1.3) were added. The protocol followed to prepare and aliquot the medium is described in Section C.2. A 50 X NaS<sub>2</sub> stock solution (Section C.1.4) was added just prior to inoculation. All medium components and the preparation protocol can be found in Section C.

#### 3.4.2.1 Microorganism-specific additives

When grown as monocultures, additives were added specifically for each microorganism. A 50 X  $\text{Na}_2\text{SO}_4$  stock solution<sup>1</sup> was added to the Dv monoculture for a final  $\text{NaSO}_4$  concentration of 7.5 mM. Mm required the addition of a 50 X sodium pyruvate stock solution<sup>1</sup> for a final concentration of 10 mM, and a 50 X NaCl stock solution<sup>1</sup> (200 g / L stock), for a final concentration of 4 g / L. Furthermore, Mm required replacement of the headspace (ca. 80% of the total volume of the Hungate tube or serum vial) to 80/20%  $\text{H}_2/\text{CO}_2$  set to 2 bar as described in Section C.3.

#### 3.4.2.2 Cryostock production

A Mm cryostock originally made from the DSMZ collection culture and a *Desulfovibrio vulgaris* strain Hildenborough sp. s1 cryostock isolated in Großkopf et al. (2016) were grown in 125 mL and 50 mL serum vials, respectively, containing 25 mL CCM. Just prior to inoculation, the 50 X  $\text{Na}_2\text{S}$  stock solution and respective additives were added and the headspace of Mm was replaced as stated in the previous section. The Dv and Mm cultures were incubated at 37 °C in the dark for 4 and 7 days, respectively. Cryostocks were prepared as described in Section C.4.

#### 3.4.2.3 Seed cultures

Prior to the experiment, seed cultures were made for each microorganism. Cryostock vials (1 mL culture) were thawed at room temperature in the dark. The 50 X  $\text{Na}_2\text{S}$  and organism-specific stock solutions were added to glass serum vials containing 25 mL CCM medium (glass vial capacity of 50 mL and 125 mL for Dv and Mm, respectively) in the anaerobic chamber. The thawed cultures were then aseptically introduced into the serum vials using a syringe and needle. Both microorganisms were incubated at 37 °C in a static, dark incubator and cultured until they reached late-exponential phase (7 days for Mm and 4 days for Dv). The headspace of Mm monocultures was replaced with 80/20%  $\text{H}_2/\text{CO}_2$  (Section C.3).

#### 3.4.2.4 Hungate tube cultures

Hungate tubes (Chemglass Life Sciences, Vineland, NJ, USA) were filled with 4.5 mL of CCM medium as described in Section C.2. Just prior to inoculation, the 50 X  $\text{Na}_2\text{S}$  and organism-specific stock solutions were added as needed. Monocultures were inoculated with 330  $\mu\text{L}$  of the corresponding seed culture, while cocultures were inoculated with 350  $\mu\text{L}$  of each seed culture. The headspace of Mm monocultures was replaced with 80/20%  $\text{H}_2/\text{CO}_2$  (Section C.3). Hungate tubes were incubated at room temperature (ca. 21 °C) in the dark without shaking.

---

<sup>1</sup>The stock solutions were prepared as the cysteine stock solution (Section C.1.3).

### 3.4.2.5 DvMm\_O2 tubes

For the DvMm control tubes used within the container, the cultures were prepared as described in Section 3.4.2.4 with an additional step. The crimp seal and blue cap were removed within the anaerobic chamber and the tubes' mouth was sealed by a breathable membrane (Z763624, Breathe Easier sealing membrane for multiwell plates, Sigma).

## 3.4.3 Theoretical gas production calculation

### 3.4.3.1 Calculating the maximum amount of methane that can be produced

The full conversion of 30 mM lactate through fermented should yield 15 mM methane (maximum theoretical yield) according to the stoichiometry of the reaction shown in Equation 1.8. Therefore, 90 mL (0.09 L) medium contains 2.7 mmol lactate, obtaining a maximum theoretical yield of 1.35 mmol methane.

### 3.4.3.2 Calculating the maximum methane volume that can be produced

Ideal gas equation:  $P(Jm^{-3})V(m^3) = n(mol)R(JK^{-1}mol^{-1})T(K)$ , where  $R = 8.314 J K^{-1}mol^{-1}$ . The container was filled in the anaerobic chamber, which has a pressure of 1.3 atm ( $1.31723 \times 10^5 J m^{-3}$ ). However, it was then placed in the Faraday cage under atmospheric pressure at room temperature (assumed as 21 °C or 294.15 K). The maximum volume of CH<sub>4</sub> that can be produced can, therefore, be calculate assuming the container conserved the anaerobic chamber's (AC) overpressure ( $P_{AC}$ , Equation 3.1) and assuming it equilibrated with room pressure ( $P_{atm}$ , Equation 3.2).

$$V_{P_{AC}} = \left( \frac{1.35 \times 10^{-3} mol \cdot 8.314 JK^{-1} mol^{-1} \cdot 294.15 K}{1.31723 \times 10^5 J m^{-3}} \right) 10^6 mL m^{-3} = 25.06 mL \quad (3.1)$$

$$V_{P_{atm}} = \left( \frac{1.35 \times 10^{-3} mol \cdot 8.314 JK^{-1} mol^{-1} \cdot 294.15 K}{1.01325 \times 10^5 J m^{-3}} \right) 10^6 mL m^{-3} = 19.28 mL \quad (3.2)$$

These calculations are just indicative; except for the case of assuming equilibration with the atmosphere, the pressures in the vessel would change due to gas production (CO<sub>2</sub>, H<sub>2</sub> and/or CH<sub>4</sub>) and consumptions.

## 3.4.4 Electrochemical cell assembly

Twelve electrochemical cells were assembled as described in Chapter 2, Section 2.4.13. Briefly, a counter electrode (CE) and working electrode (W2) was placed in half-cell A (containing a

pipe adaptor in port A1), while a second working electrode were placed in half-cell B (refer to Figure 2.2 for an electrochemical cell schematic containing port labels). The half-cells were separated by a gasket and membrane ‘sandwich’ (Section 2.4.13.3) and tightened using bolts. A No. 9 stopper was placed in port B1. See Figure 2.8 for a schematic and a photograph of the assembled electrochemical cell and consult Table 2.2 for component details.

The hand-tighten assembled electrochemical cells were covered in foil and autoclaved (121 °C, 15 min) using a desktop autoclave (ST 19 T, Dixon, Wickford, UK). These were transferred to a biological cabinet. Once cooled, the bolts were tighten using adjustable spanners. The cells were then transferred into the anaerobic chamber (AC; MACS-MG-500 anaerobic workstation, Don Whitley Scientific, Shipley, UK) for degassing, making sure the foil was still in place and intact to prevent contamination.

### 3.4.5 Electrochemical cell inoculation

After degassing the electrochemical cells for a minimum of six days, these were prepared in anaerobic chamber for the experiment. 13.8 mL  $\text{NaS}_2$  (50 X) was added to 600 mL CCM medium and mixed (referred to as CCM- $\text{NaS}_2$ ). Four cells (to be the abiotic controls) were filled with 40 mL CCM- $\text{NaS}_2$  per half-cell. Eight cells were first inoculated with 5 mL of each seed culture into the corresponding half-cell. Mm was inoculated into half-cell A using port A3; the *modified stopper* (Section 2.4.12, Figure 2.7) connected to the CE was carefully lifted. Dv was inoculated into half-cell B using port B3 (see Chapter 2, Section 2.4.6 and Figure 2.2). Once inoculated, each half-cell was filled with 36 mL CCM- $\text{NaS}_2$  to promote mixing taking care no to touch the electrochemical cell or any of its components. Finally, port A3 was closed using a sterile and degassed No. 7 stopper.

### 3.4.6 Electrochemical system

Once inoculated, the cells were placed in the containers as specified in Table 3.6. Each cell electrode was connected to the corresponding container connector using the BSs soldered to wires, referred to as *wired BS*, which were described in Chapter 2, Section 2.4.15.3 and Figure 2.12. A gas collection bag was connected to port A1 using a length of tubing (see Table 2.2 for component details) and three chemical anaerobic atmosphere generation sachets (Anaerogen® pack, AN0035, Oxoid, Thermo Scientific, UK) were placed inside the container. The container lid was put in place, connected to the container body and closed. The container was then removed from the anaerobic chamber and placed in the Faraday cage.

### 3.4.6.1 Summary of electrochemical cell arrangement

The summary of electrochemical cell arrangement in the containers can be seen in Table 3.8.

**Table 3.8** Electrochemical cell arrangement within containers and their assigned channel. TOP – data arranged by container and treatment (condition); BOTTOM – data arranged by channel

CONTAINER	CONDITION	CHANNEL	LABEL
I	A	7	A1
I	B	1	B1
I	nB	-	nB1
I	nB	-	nB4
II	A	5	A2
II	A	2	A3
II	B	6	B2
II	nB	-	nB2
III	A	3	A4
III	B	4	B3
III	B	8	B4
III	nB	-	nB3

CONTAINER	CONDITION	CHANNEL	LABEL
I	nB	-	nB1
II	nB	-	nB2
III	nB	-	nB3
I	nB	-	nB4
I	B	1	B1
II	A	2	A3
III	A	3	A4
III	B	4	B3
II	A	5	A2
II	B	6	B2
I	A	7	A1
III	B	8	B4

### 3.4.7 Equipment used for electrochemical measurements

The electrochemical measurements were carried out using a potentiostat (Reference 600+, Gamry Instruments, USA). In order to be able to perform measurements on multiple electrochemical cells, a multiplexer was used (MUX; ECM8 Electrochemical Multiplexer, Gamry Instruments, USA). The equipment was controlled using Gamry Framework Software Version 7.06.

### 3.4.8 Current measurement

The current between the two working electrodes was measured using the Gamry Framework software ‘experiment’ “Multiplexed Galvanic Corrosion”. In this configuration, a positive current represents flow of electrons from W1 to W2, as confirmed by (Miller et al., 2016). The MUX was configured to be connected in ZRA mode to ensure that the working electrodes were shorted

throughout the measurement after 10 s of an open circuit potential (OCP) measurement. The OCP is a measurement of the potential difference between the two electrodes when these are not connected (i.e. open circuit) (Lefrou et al., 2012). As the potential is dynamically set to 0 V, a voltage close to 0 V is expected to be measured, with 30  $\mu$ V being the minimum potential that can be measured reliably (priv. comm. with potentiostat manufacturer). The Multiplexed Galvanic Corrosion ‘experiment’ records the current every 0.5 s for 2 s and reports the average current. The electronic connections used are listed in Table 3.9. A reading of the voltage between the REF and the W Gamry leads is recorded simultaneously and independently from the current measurement. The sample period was set to 10, 5 or 1 min, according to the monitoring stage (see the Results Section 3.2.1), without initial delay or IR compensation. All data processing and analysis was done within the R environment with custom-made functions and scripts.

**Table 3.9** Electronic connections for ZRA measurements

Gamry cable colour	Gamry lead name	Container wall connection colour (type)	Electrochemical cell electrode
Blue	Working Sense (WS)	Grey (4 mm BS)	W1
Green	Working Electrode (W)	Grey (4 mm BS)	W1
White	Reference (REF)	Grey (4 mm BS) <sup>1</sup>	W1
Red	Counter Electrode (C)	Purple (4 mm BS)	W2
Orange	Counter Sense (CS)	Purple (4 mm BS)	W2
Long Black	Floating Ground (GND)	Black (4 mm BS)	–
Short Black	Chassis (Earth) Ground (GND)	Black (4 mm BS)	–
–	–	Brown (4 mm BS) (x 2)	CE

Note that the counter electrode (CE) is not connected during the current measurement. *Table modified from Gamry ECM8’s manual*

<sup>1</sup> The crocodile clip placed in the Gamry REF lead was clipped onto a gold plated banana plug (BP) stacked with either the W or WS Gamry lead

### 3.4.8.1 Current integration over time

Current is measured in amperes (A), which is defined as Coulombs (C) per second. Coulombs, in turn is a measure of the elementary charge (i.e. electron)<sup>2</sup>. Therefore, the total amount of electrons transferred during the experiment was estimated for each condition by integrating the current ( $I$ ) measured over time using the trapezoidal rule. A trapezoid has a geometrical shape with four sides, defined by its width ( $w$ ) and two heights ( $h_a$  and  $h_b$ ). When the heights are equal ( $h_a = h_b$ ), the shape would be that of a rectangle. Its area formula is  $(w \cdot (h_a + h_b))/2$ . When used in integral calculus, the area under a curve is estimated by approximating trapezoids of a conserved width along the x-axis (independent variable) and every “height” (current value) is used twice except the first ( $I_0$ ) and last ( $I_N$ ). The trapezoidal rule was hence implemented using

<sup>2</sup><https://physics.nist.gov/cuu/Units/units.html>



Equation 3.3, to obtain the number of Coulombs ( $nC$ )

$$nC = \int I = \frac{\Delta t}{2} \left( I_{t_0} + I_{t_N} + 2 \sum_{k=1}^{N-1} I_{t_k} \right) \text{ C} \quad (3.3)$$

where  $\Delta t$  is the time period between measurements,  $N$  is the total number of time points, and  $I_{t_0}$  and  $I_{t_N}$  are the first and last current measurements, respectively.  $\Delta t$  was 600, 300 and 60 seconds for the first, second and third measurement periods, respectively (see Results Section 3.2.1).

Changes in the current sign were observed, indicating change in the direction of the electron flow rather than a loss of electrons. To account for this, the absolute number of Coulombs ( $nC_{\text{abs}}$ ) was calculated by adding the absolute components. The difference between  $nC$  and  $nC_{\text{abs}}$  ( $nC_{\text{diff}}$ ) was calculated only as a measure of the discrepancy between them. In future, this could potentially be used to calculate an error.

$$nC_{\text{abs}} = \int I = \frac{\Delta t}{2} \left( \text{abs}(I_{t_0}) + \text{abs}(I_{t_N}) + 2 \sum_{k=1}^{N-1} \text{abs}(I_{t_k}) \right) \quad (3.4)$$

$$nC_{\text{diff}} = nC_{\text{abs}} - nC_{\text{abs}} \quad (3.5)$$

The relationship between the number of coulombs ( $nC$ ) and the number of electrons ( $e$ ) is shown in Equation 3.6. The maximum theoretical yield of electrons transferred through full lactate (30 mM) consumption of both half-cells is shown in Equation 3.7. This value was then translated to the maximum theoretical number of coulombs ( $nC_{\text{lac}}$ ) achievable from full lactate consumption in the electrochemical cells (90 mL culture volume; Equation 3.8), given that every mol of lactate produces 4 moles of electrons ( $e$ ).

$$1 \text{ C} = 6.242 \times 10^{18} e \quad (3.6)$$

$$e_{\text{lac}_L} = 0.03 \text{ mol} \left( \frac{4 \text{ mol } e}{1 \text{ mol lac}} \right) \left( \frac{6.0221409 \times 10^{23} e}{1 \text{ mol } e} \right) = 7.226569 \times 10^{22} e \text{ per L culture}$$

$$e_{\text{lac}} = e_{\text{lac}_L} \cdot \left( \frac{1 \text{ L culture}}{1000 \text{ mL culture}} \right) \cdot 80 \text{ mL culture} = 6.503912 \times 10^{21} e \text{ per 80 mL culture} \quad (3.7)$$

$$nC_{\text{lac}} = e_{\text{lac}} \cdot \left( \frac{1 \text{ C}}{6.242 \times 10^{18} e} \right) = 1041.96 \text{ C} \quad (3.8)$$

### 3.4.9 Electrochemical Impedance Spectroscopy (EIS) measurement

Electrochemical Impedance Spectroscopy (EIS) is an electrochemical method used to determine the resistance (impedance) of an electronic circuit<sup>3</sup>. This is achieved by applying a small alternating

current (AC) perturbation determined by its frequency and measuring the response through the circuit. The data obtained can be used to extract information about the system by using equivalent circuit models to describe it<sup>3</sup>.

Here, the Randles cell, shown in Figure 3.13, was used to describe the electrochemical cells<sup>3</sup>. This circuit models a cell where polarization is due to a combination of kinetic and diffusion processes. The Gamry Analysis software was used to fit the parameters listed below with the EIS data using the Simplex Method:

**$R_s$**  - solution or electrolyte resistance

**$R_{ct}$**  - charge transfer or polarization resistance

**$W$**  - Warburg impedance (semi-infinite diffusion; rate determining step)

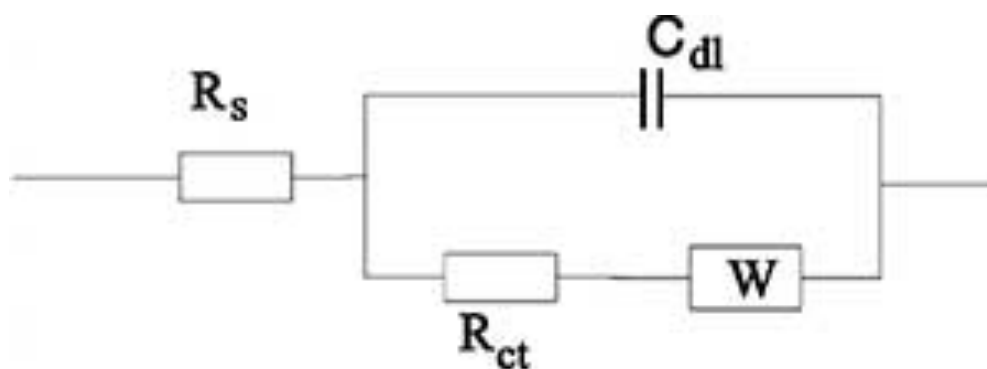
**$C_{dl}$**  - dual layer capacitance

Potentiostatic EIS was implemented with the Gamry Framework software using the frequency range between  $10^4$  and 0.01 Hz and a sinusoidal perturbation with a 10 mV amplitude at 0 V v OCP. The Gamry lead connections used for EIS are shown in Table 3.10. EIS measurements were performed on the working electrodes, connecting W1 to the Gamry W/WS leads and W2 to Gamry's C/CS leads. All data processing and analysis was done within R with custom-made functions and scripts. The OCP was measured before and after EIS to evaluate the stability of the system.

### 3.4.10 Cyclic voltammetry (CV) measurement

Cyclic voltammetry (CV) was used to investigate the electron transfer reactions present and measure the reduction potential of dominant redox reactions in the system (Marsili et al., 2008). For this measurement, a three electrode system is required to characterise each working electrode (WE)

<sup>3</sup><https://www.gamry.com/application-notes/EIS/basics-of-electrochemical-impedance-spectroscopy/>



**Figure 3.13** EIS model: the Randles cell model with double layer capacitance and Warburg impedance was used as a equivalent circuit to the physical electrochemistry measured.  **$R_s$**  is the solution resistance,  **$R_{ct}$**  is the charge transfer or polarization resistance,  **$W$**  is the Warburg impedance (semi-infinite diffusion), and  **$C_{dl}$**  is the dual layer capacitance. Figure reproduced from Gamry's application note on EIS basics<sup>3</sup>

**Table 3.10** Electronic connections for EIS measurements

Gamry cable colour	Name	Electrochemical cell electrode
Blue	Working Sense (WS)	W1
Green	Working Electrode (W)	W1
White	Reference (REF)	W2
Red	Counter Electrode (C)	W2
Orange	Counter Sense (CS)	W2
Long Black	Floating Ground (GND)	–
Short Black	Chassis (Earth) Ground (GND)	–
–	–	CE

Note that the counter electrode (CE) is not connected during the current measurement. *Table modified from Gamry ECM8's manual*

separately and the potential is measured against a reference electrode (RE) in order to be able to place it in an absolute reduction potential scale given by the standard hydrogen electrode (SHE), thus making the data comparable.

To prevent the introduction of KCl and silver particles into the medium, it was determined that the Ag/AgCl reference electrodes (RE) should be introduced at the end of the experiment to carry out the CV characterisation on the working electrodes. Each container was disconnected from the potentiostat and introduced into the anaerobic chamber. The lid was removed, the REs inserted into port B3 of the electrochemical cells and connected internally to the corresponding container connector.

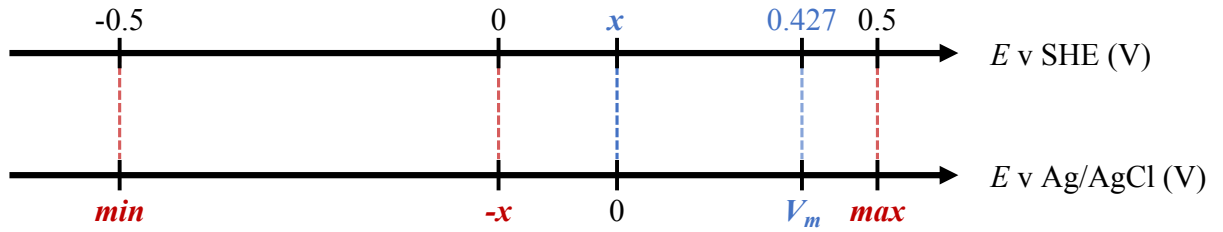
The RE had to be previously characterised, which is described in Chapter 2, Section 2.2.2, Figure 2.5. Briefly, the RE was introduced into a redox buffer and the potential between the RE and a platinum wire was measured ( $V_m$ ). The redox buffer has a potential of 0.427 V v SHE. Therefore, the translating value between reference scales,  $x$  was calculated, which represents the potential v Ag/AgCl corresponding to 0 V v SHE ( $x = 0.427 - V_m$ , Equation 2.1).

CV was implemented with the Gamry Framework software scanning from 0 to 0.5 to -0.5 to 0 V v SHE for comparison purposes using a scan rate of 100 mVs<sup>-1</sup>. The translating value  $x$  was used to calculate the equivalent potential values v Ag/AgCl for each electrochemical cell based on the RE introduced. Therefore, all CV curves were measured from  $-x$  to  $max$  (Equation 3.9) to  $min$  (Equation 3.10) back to  $-x$  V v Ag/AgCl as visually described in Figure 3.14.

$$max = x + 0.5 \quad (3.9)$$

$$min = x - 0.5 \quad (3.10)$$

The Gamry lead connections used for CV are shown in Table 3.11. All data processing and



**Figure 3.14** Method to achieve comparable CV by translating potential values across reference scales. See the main text and Equations 3.9 and 3.10.

**Table 3.11** Electronic connections for CV measurements

Gamry cable colour	Name	Electrochemical cell electrode
Blue	Working Sense (WS)	W1 or W2
Green	Working Electrode (W)	W1 or W2
White	Reference (REF)	RE
Red	Counter Electrode (C)	CE
Orange	Counter Sense (CS)	CE
Long Black	Floating Ground (GND)	–
Short Black	Chassis (Earth) Ground (GND)	–

Note that one working electrode (WE) is analysed at a time. *Table modified from Gamry ECM8's manual*

analysis was done within the R environment with custom-made functions and scripts.

#### 3.4.10.1 Mid-peak potential estimation

The mid-peak potential ( $E_{mid}$ ) was estimated using Equation 3.11

$$E_{mid} = 1/2(E_{p,c} + E_{p,a}) \quad (3.11)$$

where  $E_{p,c}$  and  $E_{p,a}$  are the cathodic and the anodic peak potentials, respectively (Scholz, 2010).  $E_{p,c}$  and  $E_{p,a}$  were estimated with function `first_derivative_inflection()` which the author developed for this work and is included below as Code 3.1 with inputs  $V$  and  $I$ .  $V$  and  $I$  are the voltage and the current, respectively, for a manually selected region of the CV trace. A region from -100 to 200 mV was manually defined to find the cathodic peak, while a region from -300 to 0 mV was defined to find the anodic peak (see Figure D.5). A rolling window with 0.5% of the data (at least 5 points) was then used to calculate the first derivative (slope) across the trace. These were stored in an array. The array containing the slopes ( $s^i$ ) was scaled ( $s^i_{scaled}$ ) by multiplying the values by the maximum current in the range ( $I_{max}$ ) and dividing by the mean slope:

$$s^i_{scaled} = s^i \cdot I_{max} / \text{mean}(s^i) \quad (3.12)$$

**Code 3.1** Peak identification using the first derivative

```

1 first_derivative_inflection <- function(V, I){
2   nRows <- length(I)           # Number of readings contained
3   window = ceiling(max(nRows*0.005,5)) # size for rolling window
4   slope <- numeric((nRows-window)) # Store slope values
5
6   # Calculate the derivative
7   for (j in 1:(nRows-window)){
8     linearModel <- lm(I[j:(j+window-1)]~V[j:(j+window-1)])
9     slope[j] <- linearModel$coefficients[[2]]
10  }
11
12  scaled_slope <- c(slope, numeric(window))*max(I)/mean(slope)
13
14  error <- min(abs(adjusted_slope-I))
15  MIN_index <- which.min(abs(adjusted_slope-I))
16
17  return(list(error=error, MIN_index=MIN_index))
18 }

```

Figure D.6 shows the region of CV trace used and the scaled slope. The peak identification by identifying a change in the sign of the slope was not successful as the change in the slope is not perpendicular to the x-axis. An alternative was to locate the intersection between the CV trace and the scaled slope. To achieve this, the current was subtracted from the scaled slope ( $s_{\text{scaled}}^i - I$ ), as can be seen in Figure D.7, along with a reference horizontal line at “ $y = 0$ ” (black) to indicate the intersection of the current and scaled slopes (where the difference between them is 0) (note that  $y$  refers to the y-axis, not a variable). The peak potential (either  $E_{p,c}$  or  $E_{p,a}$ , depending on the range of CV trace analysed) was estimated at the point where the scaled slope crossed the CV trace. This was found by calculating the minimum of the absolute of the scale slope minus the current ( $s_{\text{scaled}}^i - I$ ). This value was estimated as the error, as it expected to be large for poor fits. The voltage at which this minimum occurred was assigned as the peak potential.

$$error = \min(abs(s_{\text{scaled}}^i - I)) \quad (3.13)$$

### 3.4.11 Gas chromatography

5 mL of the gas collection bag were sampled into a 10 mL glass syringe fitted with pressure lock (Pressure-Lok, Vici, Baton Rouge, LA, USA). The pressure lock was closed after sampling. The syringe's needle was put in the sampling port for injection into the gas chromatograph. Once in place, the pressure lock was opened and ca. 1 mL gas was injected into the gas chromatograph. Gas measurements were performed on an Agilent 7890A Gas chromatograph (Agilent Technologies, Craven Arms, UK), equipped with a 6 foot 60/80 mole sieve 5A column (Supelco, Gillingham, UK) for hydrogen measurement and a 6 foot 80/100 Porapak Q column (Supelco, Gillingham,

UK) for methane measurement. 99.995% - pure methane (Thames-Restec, Saunderton, UK) or 80%/20% H<sub>2</sub>/CO<sub>2</sub> mixture (BOC, Coventry, UK) were used as standard gases for calibration of methane and hydrogen measurements, respectively.

### **3.4.12 Ion chromatography**

Ion Chromatograph Dionex (ICS-5000+ DP, Dionex, Thermo Scientific, USA) was used with an anion—column with 4 µm ion exchange matrix beads (Dionex IonPac AS11-HC-4µm (2 x 250 mm) RFIC & HPIC, Dionex, Thermo Scientific, USA) using a 1.5 mM KOH solution as the eluent. The typical pressure and downstream conductivity ranges are ca. 4,300 psi and 0.5 – 0.6 µS, respectively.

#### **3.4.12.1 Calibration standard solution preparation**

The CCM medium solution (Section 3.4.2) was prepared without lactate, without degassing and without Na<sub>2</sub>S. 60 mM lactate and acetate solutions were prepared in the CCM solution and Na<sub>2</sub>S stock solution was added. Both solutions were then diluted to achieve concentrations of 40 – 5 mM in steps of 5 mM and then 2.5 – 0.5 mM in steps of 0.5 mM using the CCM medium with Na<sub>2</sub>S.

The calibration standard solutions were filtered through a 0.22 µm polyamide (nylon) contained in a spin cartridge (8169, Costar Spin-X Centrifuge Tube Filters 0.22 µm Pore NY Membrane Nonsterile, Corning B.V Life Sciences) by centrifuging them at 13,500 rpm for 2 min in a bench-top centrifuge (SCF2, Stuart). Finally, each solution was diluted 100-fold using ultrapure water into sample vials (079812, Thermo Fisher).

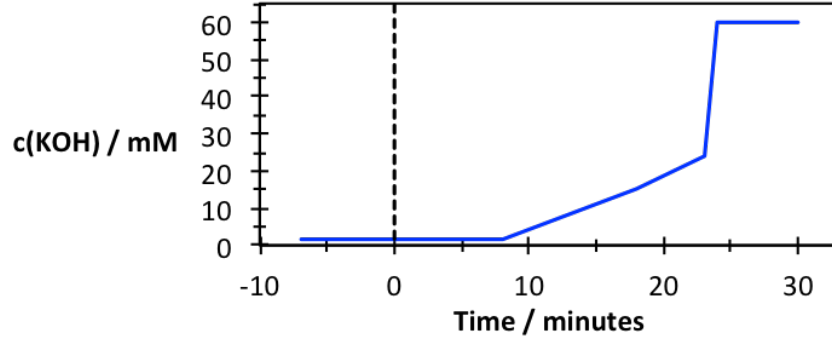
#### **3.4.12.2 Sample preparation**

All samples were filtered through a 0.22 µm polyamide (nylon) contained in a spin cartridge by centrifuging them at 13,500 rpm for 2 min in a bench-top centrifuge and stored at -20 °C. On the day of analysis, the samples were thawed at room temperature and centrifuged at 13,500 rpm for 1 min for any solids to be precipitated and not introduced into the equipment. The samples were then diluted 100-fold using ultrapure water in sample vials.

#### **3.4.12.3 Separation protocol**

The flow rate was set to 0.38 mL min<sup>-1</sup> and the column temperature to 30 °C. A gradient (Figure 3.15) was used for separation as follows:

-7 - 0 min	pre-run (equilibration) 1.5 mM KOH
0 - 8 min	1.5 mM KOH (isocratic)
8 - 18 min	increase to 15 mM KOH
18 - 23 min	increase to 24 mM KOH
23 - 24 min	increase to 60 mM KOH
24 - 30 min	stay at 60 mM KOH



**Figure 3.15** Gradient used for ion separation.

#### 3.4.12.4 Calibration

A linear regression of the peak area ( $A$ ) against the known concentrations ( $c$ ) was implemented with the `lm()` function in the *stats* package in R. The equations found are shown in Equations 3.14 and 3.15 for lactate and acetate, respectively. This was then used to calculate the concentration of the unknown samples. Figures D.11A and D.11B show graphs of the peak area against the concentration, including the values for the standards used, the linear regression line and the fitted data (unknown samples). Extrapolation was required in both cases due to unforeseen high concentrations measured in the unknown samples.

$$A_{lac} = 0.029166 \cdot c + 0.0025401 \times 10^{-5} \quad (R^2 = 0.96096) \quad (3.14)$$

$$A_{ace} = 0.0256 \cdot c - 1.1014 \times 10^{-5} \quad (R^2 = 0.96096) \quad (3.15)$$

$$\Delta c = c_{T_{end}} - c_{T_0} \quad (3.16)$$

#### 3.4.13 Optical density (OD<sub>600</sub>)

The optical density (OD) of Hungate tube cultures was measured at 600 nm in a Spectronic 200 photometer fitted with test-tube holder (Thermo Scientific, Waltham, MA, USA). Un-inoculated medium from the same batch was used as blank, and raw reads were corrected for the Hungate tube thickness to result in a 1 cm path length according to Equation 3.17.

$$OD_{600} = \frac{OD_{\text{measured}} - \text{blank}}{1.8} \quad (3.17)$$

### 3.4.14 Statistical analyses

All statistical analyses were implemented using the *stats* package in R to 95% confidence levels. A two sided Student's t-Test was applied to determine whether there was a significant difference between the mean of two different groups and was implemented with the `t.test( )` function. An F-test was carried out prior to the Student's t-Test (implemented with the `var.test( )` function), to determine whether the variance between the two samples was equal. The outcome of this test was used to determine the *var.equal* argument in the `t.test( )` function. When appropriate, a paired t-Test was implemented with the same function, but with the argument *paired* set to *TRUE* (*var.equal* argument unused).

When more than two groups were compared, a one-way analysis of variance (ANOVA) was used to test whether the mean value of a given variable were equal across the different groups or treatments. It was implemented using the `anova( )` function. When the ANOVA resulted in a significant difference (i.e. the mean values were not equal for all the groups or treatments), a Post Hoc test was carried out, implemented with a Tukey Honest Significant Differences test and the `TukeyHSD( )` function, to determine which groups or treatments were significantly different.



# Chapter 4

## MetQy

### *An R package to query metabolic functions of genes and genomes*

*The work presented in this chapter has been published in Bioinformatics as an Application Note (Martinez-Vernon et al., 2018), included in Appendix E.1. The source code, documentation and wiki can be found in the GitHub repository <https://github.com/OSS-Lab/MetQy/>. The package documentation can also be found in Appendix E.2. In the week following its online availability (05/06/2018), the GitHub repository was viewed 355 times by 57 visitors. As of 23/08/2018, it is being watched by 5 accounts and it has been starred 7 times and forked twice. MetQy is a free software for academic purposes, but not for commercial use as stated in the licence (<https://github.com/OSS-Lab/MetQy/blob/master/LICENCE>). As this licence is not compliant to the list accepted by CRAN, the R package repository, MetQy was not submitted to be published as part of that repository.*

#### 4.1 Introduction

The advent of molecular biology in the genomic era has made the characterization and analysis of genomic sequences a key part of all areas of life sciences research. In the case of single-cell organisms, identification of specific functions within the genome directly influences our ability to assess their reproductive fitness in a given environment and their potential roles in ecological and biotechnological settings. As genes encode for proteins, and interactions among proteins underpin metabolic and signalling pathways, we should theoretically be able to translate genomic data into physiological predictions. Such predictions are highly relevant for many disciplines ranging from microbial ecology and evolution to metabolic engineering.

Genomic databases are a prerequisite for making these predictions, but their full use also

requires computational tools that allow easy access and systematic analysis of the data. Interestingly, such “enabling” computational tools are not common and most researchers are limited in their analysis to the user interfaces available on genomic databases. These databases aim to feature user-friendly query and retrieval interfaces. However, as the databases grow, these interfaces inevitably become difficult to use for systematically analysing the available data in an automated fashion. Furthermore, even well-designed user interfaces are limiting in the sense that they are not open-source and do not necessarily allow development of new types of analysis or systematic queries on the available data. To the author’s knowledge, there are no tools that facilitate the analysis and information retrieval of the relationship between genomic data and biological function. The option of downloading entire databases and developing computational analysis tools on them remains a niche expertise that is still not available in many research labs.

The Kyoto Encyclopedia of Genes and Genomes (KEGG) database is one of the oldest and most comprehensive databases. Its primary aim over the last couple of decades has been the digitising of the current knowledge on genes and molecules and their interactions (Kanehisa, 1997; Kanehisa and Goto, 2000) and it includes 16 databases and 3 sequence data collections (Kanehisa et al., 2017). While these data can be analysed via different tools on the KEGG website, the existing web interface allows only specific retrieval of information and analysis. Furthermore, although the whole of the data can be downloaded via (paid) FTP access, the systematic analysis of these data in a user-defined manner remains difficult. That being said, there have been research groups with the expertise required to take advantage of the wealth of information contained within KEGG that have developed widely used computational tools that perform different types of analysis. For instance, PICRUSt was developed to predict the functional content of a metagenome based on a marker gene (e.g. 16S rRNA) of full genomes (Langille et al., 2013), while BlastKOALA and GhostKOALA were developed by the KEGG team as genome annotation tools (Kanehisa et al., 2016b). Nevertheless, there are no tools to our knowledge that facilitate the analysis and information retrieval of the relationship between genomic data and biological function. Therefore, **MetQy**, an open-source, easy-to-use and readily expandable R package was developed for such analysis and queries.

**MetQy** relies mainly on three KEGG databases for analysing physiological functions: KEGG orthology, KEGG module and KEGG genome. **MetQy** was developed to readily interface between these three key databases and perform automated cross-analysis on them. This package has the functionality to convert these KEGG database files into R data frames and introduces a set of functions that allow querying genes, enzymes and functional modules across genomes and vice versa. **MetQy** contains extensive documentation for each function and provides usage examples. **MetQy** uses the R-platform, a programming language and environment for statistical computing

and graphics, which is a highly extensible Free Software<sup>1</sup>. R is commonly used among biologists, featured in undergraduate education and contains extensive statistical packages (Carson and Basiliko, 2016). Moreover, MetQy allows easy and systematic analysis of metabolic and physiological functions within genomes and metagenomes for the microbial ecology, evolution, systems and synthetic biology research communities. Furthermore, it will enable better understanding of genotype–phenotype mapping in single-celled organisms and provide guidance in cellular engineering through synthetic biology.

## 4.2 MetQy and the use of KEGG data

As mentioned before, even though KEGG contains at least 16 databases (Kanehisa et al., 2017), MetQy relies mainly on three KEGG databases for analysing physiological functions: KEGG orthology, KEGG module and KEGG genome, which are briefly described in the next sections. In addition to these, the database KEGG enzyme also contains valuable information and is therefore briefly described next. MetQy contains in-built KEGG data (downloaded 20/02/2018) which is hidden from the user, in compliance with the KEGG FTP licence. Users with FTP access can use the parsing functions (Section 4.3.1) to process the KEGG database files and to provide up-to-date information to the query functions (Section 4.3.2). MetQy includes the data entries listed in Table 4.1.

**Table 4.1** KEGG databases in MetQy

DATABASE	NUMBER OF ENTRIES	NOTES
KEGG orthology	21,800	–
KEGG genome	5,244	Genomes without annotations were removed. Genomes <i>prn</i> (T04692) and <i>con</i> (T04096) are not included due to limitations of the Windows OS folder naming convention.
KEGG enzyme	6,087	–
KEGG module	780	Modules M00611 to M00618 have been removed, as these have KEGG module definitions that involve other modules.

### 4.2.1 KEGG orthology data

Orthologs refer to genes that are derived from the same ancestor gene and, therefore, their sequences are similar (Alberts et al., 2002). KEGG orthology<sup>2</sup> is a database of molecular functions represented in terms of functional orthologs. KEGG orthologs are manually defined in the context of KEGG networks. A functional ortholog is manually defined in the context of KEGG molecular networks, (namely, KEGG pathway maps, BRITE hierarchies and KEGG modules) and are iden-

<sup>1</sup><https://www.r-project.org/about.html>

<sup>2</sup><https://www.genome.jp/kegg/annotation/ko.html>

tified by unique K numbers. Experimentally characterised genes and proteins in specific organisms have been used to identify orthologs in other organisms (Kanehisa et al. (2016a) and (2016b)).

### 4.2.2 KEGG enzyme data

The KEGG enzyme database<sup>3</sup> is based on the ExplorEnz database at Trinity College Dublin (McDonald et al., 2009). ExplorEnz is a database containing the enzyme nomenclature and classification system, the Enzyme Commission (EC) number system, developed by two biochemical nomenclature committees<sup>4</sup>, IUPAC<sup>5</sup>–IUBMB<sup>6</sup> Joint Commission on Biochemical Nomenclature (JCBN) and the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). The EC number system was developed to name and classify the reactions catalysed by the enzymes and is not concerned by their protein structures. As EC numbers do not specify enzymes *per se*, but enzyme-catalysed reactions, enzymes catalysing the same reaction receive the same EC number. Note, however, that the reactions associated with EC numbers are often generic (i.e. they provide groups of chemicals as substrate/product, rather than a specific compound) and a single EC number can be described by closely related reactions. The EC numbers in the KEGG enzyme database can, therefore, be mapped to one or more K numbers and one or more organisms.

The EC number nomenclature consists of 4 numerical positions separated by periods (e.g. “1.10.3.9” or “6.5.1.3”). The first position refers to the enzyme class and can be one of 6:

**EC 1** – Oxidoreductases

**EC 2** – Transferases

**EC 3** – Hydrolases

**EC 4** – Lyases

**EC 5** – Isomerases

**EC 6** – Ligases

The remaining positions provide information on the sub- and sub-subclasses, which change according to the enzyme class and subclass. A list of all the EC numbers including the classes, subclasses and sub-subclasses (over 300) can be found at <http://www.enzyme-database.org/class.php>.

### 4.2.3 KEGG genome data

KEGG genome<sup>7</sup> is a repository of complete genomes identified by a unique T number (e.g. ‘T00001’) and by a 3–4 letter code (e.g. ‘eco’) (Kanehisa et al., 2017). The current entries in the KEGG genomes have been either manually annotated or have been annotated using KOALA

---

<sup>3</sup><https://www.genome.jp/kegg/annotation/enzyme.html>

<sup>4</sup><http://www.sbcs.qmul.ac.uk/iupac/jcbn/>

<sup>5</sup>IUPAC, International Union of Pure and Applied Chemistry

<sup>6</sup>IUBMB, International Union of Biochemistry and Molecular Biology

<sup>7</sup><https://www.kegg.jp/kegg/genome.html>

**Table 4.2** Summary of KEGG genome data in MetQy

KINGDOM	Number of genomes
Eukaryota	434
Bacteria	4548
Archaea	262

(KEGG Orthology And Links Annotation), KEGG’s internal annotation tool for K number assignment using SSEARCH computation, based on the amino acid sequences (Kanehisa et al., 2016b). 99.9 % of the annotated genomes in KEGG come from the RefSeq (Pruitt et al., 2002) and GenBank (Benson et al., 2012) databases. Table 4.2 lists the number of genomes by Kingdom contained in MetQy.

At the start of the KEGG project, genome annotations were carried out by assigning EC numbers to the genome sequences (Bono et al., 1998; Kanehisa and Goto, 2000). This limited the matching procedure between genomic and network information to enzyme–catalysed reactions. Therefore, K numbers were introduced into KEGG in 2004 to extend the matching procedure to include regulatory pathways and to overcome various problems inherent in the enzyme nomenclature (Kanehisa, 2004). A mapping between K and EC numbers has been established within KEGG and, therefore, the EC numbers are still included within the KEGG genome database. Hence, KEGG genomes have both K numbers and EC numbers associated with them. However, as stated in Section 4.2.2, it is worth keeping in mind that one EC number may represent multiple reactions and multiple enzymes sequence families. Additionally, KEGG modules have been defined in terms of K numbers, which have been mapped to EC numbers. Therefore, the use of EC numbers is less accurate when evaluating KEGG modules.

#### 4.2.4 KEGG module data

Finally, KEGG module<sup>8</sup> is an expert–curated database that groups K numbers into *modules*. These modules (identified by M numbers) are thought to represent functional units and are defined in terms of the necessary genes (i.e. K numbers) for the expression of module components. There are four types of module:

- *pathway modules* refer to functional units in KEGG metabolic pathway maps,
- *structural complexes* refer to molecular machines or complexes,
- *functional sets* describe other essential sets, and
- *signature modules* are groups of genes associated with a phenotype.

---

<sup>8</sup><https://www.kegg.jp/kegg/module.html>

Examples of modules are those for the TCA cycle, nitrogen assimilation or methane oxidation (Muto et al., 2013; Kanehisa, 2013; Kanehisa et al., 2014).

#### 4.2.4.1 KEGG module definition

Each KEGG module is defined by a set of blocks. Each block consists of a logical expression of the K numbers involved (genes represented as K numbers). For example, the cysteine biosynthesis module (M00021) has two blocks, each composed of the following logical expressions:

Block 1. **K00640**

Block 2. **K01738|K12339|K13034|K17069**

The pipe (|) denotes an OR operation. In other examples, the ampersand (&) denotes an AND operation. The blocks are space-delimited and represent the reactions or molecular complexes required for the functionality represented as a module. The block-based definition of modules facilitates the evaluation of whether a genome contains a given module by assessing each module block.

KEGG Mapper<sup>9</sup> is a web-interface toolset that allows the evaluation of KEGG modules given a genome as a list of genes (Kanehisa et al., 2012). The tool returns a list of the modules that are complete and that have one block missing. However, KEGG Mapper does not allow the user to define a subset of modules to be evaluated and, furthermore, it does not allow bulk analyses, requiring the user to manually enter a single list of genes at a time.

### 4.3 MetQy package description

MetQy was developed as an open-source, easy-to-use and readily expandable R package, to enable systematic analysis of KEGG data in a user-defined manner, and to facilitate information retrieval of the relationship between genomic data and biological function, as well as downstream processes and analyses.

To this purpose, MetQy consists of five families of user-called functions. A typical workflow would follow the next steps. First, KEGG FTP users would parse and format the KEGG databases for ease of use within the R environment using the MetQy *parsing* functions. Next (or non-FTP users starting point), information about the relationship between genomic data and biological function would be retrieved in a flexible manner using the MetQy *query* functions. KEGG FTP users could use the formatted data obtained previously to retrieve the most up-to-date information. Non-FTP users would benefit from the in-built KEGG data contained within the MetQy

---

<sup>9</sup><https://www.genome.jp/kegg/mapper.html>

pacakge. Finally, the *MetQy analysis & visualisation* functions would help the user carry out some analyses with the information retrieved, as well as visualising it.

The main four families of function (*parsing, query, analysis & visualisation*) are described below and example code has been included to show and facilitate their usage. The fifth family of functions comprises miscellaneous functions (named with prefix *misc.*). As these are non-essential to the package usability, they have not been included here. However, their descriptions can be found in the package documentation in Appendix E.2 and at [https://github.com/OSS-Lab/MetQy/blob/master/MetQy\\_1.1.0.pdf](https://github.com/OSS-Lab/MetQy/blob/master/MetQy_1.1.0.pdf).

The coded examples shown in the next sections can be run within the R environment. Basic knowledge of R is assumed for a full understanding, but the examples provided are meant to be copy-pasted onto the console and executed as-is in the order provided. Note that the text copied might need formatting, especially underscores, quote marks and hyphens, and that the examples can also be found in the package's GitHub wiki pages (<https://github.com/OSS-Lab/MetQy/wiki/>).

The examples simulate the console. Commands to be run are preceded by '>', **comments begin with a hash symbol ('#') and are in green**, and lines without either of these simulate printed output onto the screen. **Functions and keywords are highlighted in blue** and **text or strings are in magenta**. Inline code (including variable names) in the main text is shown as such: *this is inline code*. MetQy functions are shown as italicised code, but without brackets for simplicity (e.g. *parseKEGG.file*, rather than *parseKEGG.file( )*). Throughout the usage examples, it is assumed that the MetQy package has been installed and loaded by calling `library(MetQy)` before running the examples. Instructions on how to install MetQy can be found in Code 4.1. Alternative installation steps can be found at <https://github.com/OSS-Lab/MetQy/wiki/About-MetQy-and-installation>. Some of the coded examples use data contained within the MetQy package as examples of input for the different functions. These objects are retrieved by using the `data( )` R function and are named with the prefix *data\_* (e.g. 'data\_example\_KOnumbers.vector').

#### Code 4.1 MetQy installation steps

```
1 ## Install the 'devtools' package if not in library
2 > if(length(grep("devtools", installed.packages()[,1]))==0) install.packages("
   devtools")
3
4 ## Install MetQy – required dependent packages are installed automatically
5 > devtools::install_github('OSS-Lab/MetQy', subdir='MetQy-1.1.0', dependencies=TRUE)
```

### 4.3.1 MetQy parsing functions

The MetQy *parsing* functions allow users with current KEGG FTP access to increase the usability of the KEGG databases and to provide up-to-date data to the *query* family of functions. As per KEGG’s license, these data are hidden from the user and can only be accessed by MetQy functions, allowing direct usage of MetQy as downloaded.

There are two main file types in the KEGG databases: flat database entry files named after the database (without extension) or files containing the relationship between databases, which are called after the databases described and have a “.list” extension. MetQy features two generic parsing functions that parse these two main KEGG file types, *parseKEGG\_file* and *parseKEGG\_file.list*, as well as file-specific functions that use these two generic functions.

#### 4.3.1.1 *parseKEGG\_file*

The database files (e.g. “module”) present entries sequentially by having information fields identifiable by being in capital letters (e.g. “ENTRY” and “NAME”). The information fields vary across databases. *parseKEGG\_file* automatically detects the fields of the KEGG data and transforms these into variables, stored in an R data frame.

MetQy contains six KEGG database-specific functions that use *parseKEGG\_file* to generate R data frames followed by a series of data formatting steps specific to the KEGG database. These functions are called *parseKEGG\_* followed by the specific KEGG database name:

- *parseKEGG\_compound*
- *parseKEGG\_genome*
- *parseKEGG\_module*
- *parseKEGG\_enzyme*
- *parseKEGG\_ko*
- *parseKEGG\_reaction*

There is also an umbrella function, *parseKEGG\_execute\_all*, that allows automatic execution of these individual parsing functions (and those described in the next section). A usage example is presented in Code 4.2.

#### 4.3.1.2 *parseKEGG\_file.list*

*parseKEGG\_file.list* transforms the file containing the relationship between two KEGG database entries into a binary matrix, where a **1** indicates the relationship between the two entries and **0** means no relationship. For example, the mapping between K numbers and EC numbers is contained in the *ko\_enzyme.list* file and shows which K numbers correspond to which EC numbers.

MetQy contains two KEGG file-specific functions that use *parseKEGG\_file.list* to generate R data frames. These functions are called *parseKEGG* followed by the specific KEGG file name (not including extension):



**Code 4.2** Usage example for *parseKEGG\_compound*

```
1
2 # The parent folder should contain the following (KEGG FTP structure):
3 # brite/
4 # genes/
5 # ligand/
6 # medicus/
7 # module/
8 # pathway/
9 # README.kegg
10 # RELEASE
11 # xml/
12 > compound_reference_table <- parseKEGG_compound(KEGG_path)
13 # A .txt file (tab separated) is written to 'output/' (relative to the current
    working directory)
```

- *parseKEGG\_ko\_enzyme* (relationship between KEGG orthologs and EC numbers)
- *parseKEGG\_ko\_reaction* (relationship between KEGG orthologs and KEGG reactions)

There is also an umbrella function, *parseKEGG\_execute\_all*, that allows automatic execution of these two parsing functions along with the ones mentioned before. A usage example is presented in Code 4.3.

**Code 4.3** Usage example for *parseKEGG\_ko\_enzyme*

```
1 > ko_enzyme_map <- parseKEGG_ko_enzyme(KEGG_path)
2 # A .txt file (tab separated) is written to 'output/' (relative to the current
    working directory)
```

### 4.3.2 MetQy query functions

The *MetQy query* family of functions allow the user to query the KEGG data structures in a systematic (and automated) way. These functions rely on built-in, formatted KEGG data (downloaded in February 2018), which is part of the *MetQy* package and which is not directly accessible by the user. However, the *MetQy query* functions feature optional arguments that allow users to provide up-to-date data (by using the *MetQy parsing* functions on KEGG FTP data) or their own data structures. Within user-provided data, advanced users can also define and evaluate custom-made KEGG-style modules. Additional *query* functions can be readily developed by the users, allowing expansion of *MetQy*.

*MetQy* features five query functions for key functional analyses which are described below.

- *query\_genomes\_to\_modules*
- *query\_genes\_to\_genomes*
- *query\_modules\_to\_genomes*
- *query\_genes\_to\_modules*
- *query\_missingGenes\_from\_module*

**Code 4.4** Usage example for *query\_genomes\_to\_modules* – input: organisms’ names

```
1  ## USE ORGANISMS' NAMES
2  > NAMES      <- c("escherichia coli","heliobacter") # partial names
3  > OUT        <- query_genomes_to_modules(NAMES,
4                                     MODULE.ID = paste("M0000",1:5,sep=""))
5  > dim(OUT$MATRIX) # rows cols
6  [1] 66  5
7  # 66 organisms matched the partial names provided; (we specified 5 modules)
```

#### 4.3.2.1 *query\_genomes\_to\_modules*

*query\_genomes\_to\_modules* evaluates the module coverage of a set of genes (K numbers) or genomes. The module completeness, referred as *module completeness fraction* (*mcf*), is calculated for each module as the number of complete blocks divided by the total number of blocks and is thus reported as a fraction (see Section 4.2.4). A genome or gene set containing all the required genes would result in a *mcf* of 1. The *mcf* calculations is carried out by calling MetQy’s functions *misc\_geneVector\_module()* and *misc\_evaluate\_block()*.

*query\_genomes\_to\_modules*’s input consists of specifying KEGG module(s) using M number(s) to evaluate across given genome(s). The genomes can be specified by either (1) KEGG genome identifier(s) (T number(s) or 3–4 letter code(s); Code 4.6), (2) organism name(s) (case insensitive; Code 4.4), or (3) set(s) of genes using either K or EC numbers (Code 4.5). See Section 4.2.3 for details on the KEGG genome database. In the first two cases, the built-in KEGG data are used to retrieve the gene content for the genome(s). Note that specifying the organism name may result in multiple genomes being matched and analyzed automatically. Additionally, using EC numbers to specify the gene set is less accurate, as KEGG modules are defined in terms of KEGG orthologs and not all KEGG orthologs are mapped to EC numbers.

The main output of this function constitutes a matrix of the genome identifier(s) (rows) and the module ID(s) (columns) matching the query containing the *mcf* (OUT\$MATRIX, Code 4.7).

**Code 4.5** Usage example for *query\_genomes\_to\_modules* – input: gene sets

```
1  ##### USE USER-SPECIFIED GENE SETS
2  > data(data_example_multi_EC_KOs) # load example data set – note that this entry
   does not exist in KEGG
3  > data_example_multi_EC_KOs[1,] # trimmed for display purposes
4  ID ORG_ID ORGANISM KOs ECs
5  T09999 aa A K00013;K00014;... 1.1.1.1;1.1.1.100;...
6
7  ## USE KEGG ORTHOLOGS
8  > OUT      <- query_genomes_to_modules(data_example_multi_EC_KOs,
9                                     GENOME.ID.COL = "ID", GENES.COL = "KOs",
10                                    MODULE.ID = paste("M0000",1:5,sep=""))
11
12  ## USE EC NUMBERS
13  > OUT      <- query_genomes_to_modules(data_example_multi_EC_KOs,
14                                     GENOME.ID.COL = "ID", GENES.COL = "ECs",
15                                    MODULE.ID = paste("M0000",1:5,sep=""))
```

**Code 4.6** Usage example for *query\_genomes\_to\_modules* – input: T numbers

```

1  ## USE T numbers
2  > T.NUMEBERS <- paste("T0000", 1:5, sep="")
3  > OUT <- query_genomes_to_modules(T.NUMEBERS,
4                                     MODULE.ID = paste("M0000", 1:5, sep=""),
5                                     META.OUT = T, ADD.OUT = T)

```

**Code 4.7** Usage example for *query\_genomes\_to\_modules* – input: gene sets

```

1  > names(OUT) #"MATRIX" "QUERIES" "METADATA" "ADD_INFO" "GENOME_INFO_DATA"
2
3  ## RETRIEVE THE mcf
4  > mcf <- OUT$MATRIX
5  > mcf[1:2,]
6
7  T00001 0.8888889 1 1.0000000 0.8571429 1
8  T00002 0.8888889 1 0.7142857 0.5714286 1

```

The genome information is returned in the output `OUT$GENOME_INFO_DATA` (Code 4.8). Additional information about the KEGG modules, such as their classification, can be retrieved by using the `META_OUT` argument (Code 4.9). A data frame containing the *mcf* for every module and every genome/gene set, as well as the module name and the number of blocks in its definition, can be retrieved with the `ADD_OUT` argument (Code 4.10).

While the implementation of the *query\_genomes\_to\_modules* function is similar to KEGG mapper (refer to Section 4.2.4), there are several key features that are different. First, the KEGG Mapper’s web interface does not allow for module-specific evaluation nor for automation of the analysis. Our implementation allows for specific KEGG modules to be evaluated, given their ID, name and/or class. It also provides the capacity to determine the *mcf* of a module, rather than only identifying modules that are complete or that have one block missing. Finally, as EC numbers are widely used in systems biology, we used the KEGG orthology to translate the K number-based module definitions to EC number-based module definitions. This allows for module evaluation based on both K and EC numbers.

**Code 4.8** Usage example for *query\_genomes\_to\_modules* – input: T numbers

```

1  # Retrieve genome information (relationship between T identifier, letter code
2  # identifier and name)
3  > OUT$GENOME_INFO_DATA
4
5  ID  ORG_ID  ORGANISM
6  T00001  hin  Haemophilus influenzae Rd KW20 (serotype d)
7  T00002  mge  Mycoplasma genitalium G37
8  T00003  mja  Methanocaldococcus jannaschii DSM 2661
9  T00004  syn  Synechocystis sp. PCC 6803
10 T00005  sce  Saccharomyces cerevisiae S288c

```

**Code 4.9** Usage example for *query\_genomes\_to\_modules* – output details

```

1  ## META DATA
2  # Retrieve the module information (trimmed <...> for display purposes, only
   showing 6 of 8 columns)
3  > OUT$METADATA[1,1:3] # MODULE NAME
4  MODULE_ID    MODULE_NAME    NAME_SHORT
5  M00001      Glycolysis ...  Glycolysis – EM pathway
6
7  > OUT$METADATA[1,c(1,4:5)] # MODULE CLASSES
8  MODULE_ID    CLASS_I    CLASS_II
9  M00001      Pathway module  Carbohydrate and lipid metabolism
10
11 > OUT$METADATA$CLASS_III[1]
12 [1] Central carbohydrate metabolism

```

**Code 4.10** Usage example for *query\_genomes\_to\_modules* – input: T numbers

```

1  ## ADDITIONAL OUTPUT
2  # Retrieve additional information on the mcf (FRACTION) and the number of blocks
   in the KEGG module definition
3  > OUT$ADD.INFO[1, 1:5]
4  GENOME_ID    MODULE_ID    NAME_SHORT    FRACTION    nBLOCKS
5  T00001      M00001      Glycolysis – EM pathway  0.8888889    9

```

**4.3.2.2 query\_modules\_to\_genomes**

*query\_modules\_to\_genome* determines the KEGG genome(s) that have user-specified module(s) that are complete above a *mcf* threshold (defaults to 1, i.e. complete). The output is a matrix with the module ID(s) (columns) and KEGG genome T number(s) (rows) containing the corresponding *mcf*. A usage example is presented in Code 4.11.

**Code 4.11** Usage example for *query\_modules\_to\_genomes*

```

1  ## Single module using threshold of 1 (default)
2  > genomes <- query_modules_to_genomes("M00001")
3  > genomes[1:5]
4  T00003 T00005 T00007 T00010 T00014
5  1      1      1      1      1
6
7  ## Multiple modules using set threshold (0.75 in this case)
8  > genomes <- query_modules_to_genomes(MODULE_ID = c("M00001", "M00002"),
9  threshold = 0.9)
10 > head(genomes)

```

**4.3.2.3 query\_genes\_to\_modules**

*query\_genes\_to\_modules* determines those KEGG modules that feature specific user-specified gene(s), defined by their KEGG ortholog (K number) or Enzyme Commission (EC) identifier. The output is a data frame containing the modules (and their description) that contain the specified gene. A usage example is presented in Code 4.12.

**Code 4.12** Usage example for *query\_genes\_to\_modules*

```

1 ## Find the module ID(s) where the given KEGG ortholog or EC number are involved.
2 > modules <- query_genes_to_modules("K00844")
3 > modules
4      M00001 M00549
5 K00844      1      1
6
7 > modules <- query_genes_to_modules("1.1.1.1")

```

**4.3.2.4** *query\_genes\_to\_genomes*

*query\_genes\_to\_genomes* determines which KEGG genomes contain user-specified gene(s). The output is a binary matrix showing the presence/absence of each of the user-specified gene(s) (in rows) against the user-specified genome(s) (in columns). A usage example is presented in Code 4.13.

**Code 4.13** Usage example for *query\_genes\_to\_genomes*

```

1 ## Find the genomes that have been annotated with a KEGG ortholog
2 > genomes <- query_genes_to_genomes("K00844")
3 > genomes[,1:5]
4      T00005 T00016 T00019 T00030 T00041
5 K00844      1      1      1      1      1
6
7 ## Find the genomes that have been annotated with an EC number
8 > genomes <- query_genes_to_genomes("1.1.1.1")
9 > genomes[,1:5]
10     T00001 T00004 T00005 T00006 T00007
11 1.1.1.1      1      1      1      1      1
12
13 ## Find the genomes that have been annotated with a KEGG ortholog
14 > genomes <- query_genes_to_genomes(genes = paste("K0000",1:3,sep=""))
15 > genomes[,1:5]
16     T00003 T00004 T00005 T00008 T00009
17 K00001      0      0      0      0      0
18 K00002      0      1      1      0      0
19 K00003      1      1      1      1      1
20 # Boolean relationship, where 1 indicates that the genome (column) has been
    annotated with that KEGG ortholog (row) and 0 means that has not been.

```

**4.3.2.5** *query\_missingGenes\_from\_module*

Given a genome (or set of genes), *query\_missingGenes\_from\_module* determines the missing gene(s) (K or EC numbers) that would be required to have a complete KEGG module within that genome (or gene set). The user input to the function constitutes of either a KEGG genome identifier (T number or 3–4 letter code) or a set of genes (either K numbers or EC numbers). If a KEGG genome identifier is provided, the built-in KEGG data are used to retrieve the gene content for that genome. The function output consists of a data frame with (1) the KEGG module structure highlighting the missing genes within each block with an asterisk, (2) the absence or presence of

**Code 4.14** Usage example for *query\_missingGenes\_from\_module*

```
1 > data(data_example_KOnumbers_vector) # Load data
2 > OUT <- query_missingGenes_from_module(data_example_KOnumbers_vector, "M00010",
3                                         PRINT_TO_SCREEN = F)
4 > print(OUT)
5   block_No PRESENT      BLOCK_DEF MISSING_GENES
6   1         1       1      K01647
7   2         2       1  K01681|K01682
8   3         3       1 K00031|*K00030*
9 # KEGG ortholog K00030 is missing, but the block is still complete, so there are no
10  genes missing (i.e. required for the module to be complete) in this example.
11 > OUT <- query_missingGenes_from_module(data_example_KOnumbers_vector, "M00046",
12                                         PRINT_TO_SCREEN = F)
13 > print(OUT)
14   block_No PRESENT      BLOCK_DEF MISSING_GENES
15   1         1       1 *K00207*|K17722&K17723
16   2         2       1      K01464
17   3         3       0  *K01431*|*K06016* K01431;K06016
18 # There is one block missing in this example, which could be completed by having
19   EITHER KOs K01431 OR K06016, according to the block definition.
```

each block with a binary indicator (see above for block presence/absence criteria) and (3) the list of missing genes. A usage example is presented in Code 4.14.

### 4.3.3 MetQy analysis functions

The MetQy *analysis* family of functions is designed to facilitate the analysis primarily of the output of the *query\_genomes\_to\_modules* function, which generates a matrix of *mcf* values for the genomes and modules analysed. The main focus of currently implemented analysis functions is to apply Principal Component Analysis (PCA) to the *mcf* matrix.

PCA is useful for data reduction, using the correlation between variables. The KEGG modules were expected to be correlated across genomes and hence the application of PCA to the *mcf* matrix valuable. The expectation was that only a few principal components (PCs) would be required to account for the majority of the variability of the data. Furthermore, plotting the first PCs might highlight clustering of similar data (Jolliffe, 2010).

PCA was implemented with the function `prcomp()` of the R package *stats*. `prcomp()` returns a list containing, among other elements, a matrix with the resulting rotated data, `x`, and a vector, `sdev`, of the variance captured by every PC. `x` has the same number of rows as the *mcf* matrix, corresponding to the genomes analysed, and as many columns as PCs or modules analysed. `sdev` has the same length as the columns of `x`.

The next following sections describe the three analysis functions that MetQy features:

- *analysis\_pca\_mean\_distance\_calculation*
- *analysis\_pca\_mean\_distance\_grouping*
- *analysis\_genomes\_module\_output*

The first two revolve around the application of PCA, while the third was designed to perform automated analyses and to use several visualisation functions, described in the next section (Section 4.3.4), compiling it in a `LATEX` report.

#### 4.3.3.1 *analysis\_pca\_mean\_distance\_calculation*

This function was designed to be called by *analysis\_pca\_mean\_distance\_grouping* (see next Section) to calculate the mean distance in the PC space of a group as defined by a ‘*FACTOR*’ as a proxy for within-group variance. The input would be a subset of the `x` object generated by `prcomp()` using the rows corresponding to a group’s members ( $p$ ) and using as many columns as dimensions are desired ( $n$ ). Thus, a row of `x` would contain the rotated data for a given genome for every PC and could be thought as a Euclidean vector ( $\mathbf{v}^i$ ) with as many dimensions as PCs. Therefore, given  $p$  points in a Euclidean space with  $n$  dimensions, *analysis\_pca\_mean\_distance\_calculation* calculates the mean distance between all points. Code 4.15 contains a usage example.

Formally, there are  $p$  points of the form  $\mathbf{v}$  with  $n$  dimensions (Equation 4.1).

$$\mathbf{v}^i = (v_1^i, v_2^i, \dots, v_{n-1}^i, v_n^i), \text{ where } i = 1, \dots, p \quad (4.1)$$

The total number of Euclidean distances ( $N$ ) is equal to the number of 2-combinations in the set of  $p$  points given by the binomial coefficient as shown in Equation 4.2.

$$N = \binom{p}{2} = \frac{p!}{2!(p-2)!} = \frac{p(p-1)}{2} \quad (4.2)$$

The Euclidean distance between two points ( $d_{xy}$ ) is calculated as described in Equation 4.3, where  $d_{xy}$  is an element of  $\mathbf{d} = (d_1, d_2, \dots, d_{N-1}, d_N)$ , a vector containing the individual distances.

$$d_{xy} = \sqrt{\sum_{k=1}^n (v_k^x - v_k^y)^2} \quad (4.3)$$

The mean distance between all  $p$  points ( $d_{mean}$ ) is given by the sum of the individual Euclidean distances (contained in  $\mathbf{d}$ ) divided by the total number of distances (Equation 4.4).

$$d_{mean} = \frac{\sum_{j=1}^N \mathbf{d}}{N} \quad (4.4)$$

Similar data is expected to cluster in a PC plot and have a small mean distance. Therefore, the mean distance could serve as a proxy for within group similarity or spread.

**Code 4.15** Usage example for *analysis\_pca\_mean\_distance\_calculation*

```
1 > data(data_example_moduleIDs)
2 > data(data_example_genomeIDs)
3
4 # Calculate the module completion fraction (mcf) for the genomes and modules
  contained in the data objects above.
5 > OUT      <- query_genomes_to_modules(data_example_genomeIDs ,
6                                         MODULE.ID = data_example_moduleIDs)
7 > pca <- prcomp(OUT$MATRIX)
8
9 > mean_dist <- analysis_pca_mean_distance_calculation(pca$x)
10 > print(mean_dist)
11 # [1] 0.4805169
```

#### 4.3.3.2 *analysis\_pca\_mean\_distance\_grouping*

As mentioned before, PCA could be useful to find clusters of similar data. It would be expected that genomes grouped based on their taxonomy would cluster together. The mean Euclidean distance for such groups would be expected to be small and hence could be used as a proxy for spread. *analysis\_pca\_mean\_distance\_grouping* was designed to calculate the mean Euclidean distance for the groups defined in a ‘*FACTOR*’ (vector or list) as a proxy for within-group variance. This function takes in the ‘*FACTOR*’, such as a vector of the genomes’ genus, and the rotated data object (x) resulting from applying PCA on the *mcf* matrix (implemented with the *stats::prcomp* function). *analysis\_pca\_mean\_distance\_grouping* calls the function described above, *analysis\_pca\_mean\_distance\_calculation*, to calculate the mean Euclidean distance for each group. In addition, it generates a scatter plot of the first two PCs with the groups overlaid (by calling *plot\_scatter*) and has the option of plotting the mean distance for each group (by calling *plot\_scatter\_byFactors*). A

**Code 4.16** Usage example for *analysis\_pca\_mean\_distance\_grouping*

```
1 ## LOAD EXAMPLE DATA
2 > data(data_example_moduleIDs)
3 > data(data_example_genomeIDs)
4
5 ## Calculate the module completion fraction (mcf) for the genomes and modules
  contained in the data objects above.
6 > OUT      <- query_genomes_to_modules(data_example_genomeIDs ,
7                                         MODULE.ID = data_example_moduleIDs)
8 > pca <- prcomp(OUT$MATRIX)
9
10 ## Group data
11 > organisms <- OUT$GENOME.INFO.DATA$ORGANISM
12 > organisms[1:2]
13 [1] "Methanobacterium curvum Buetzberg" "Methanobrevibacter millerae SM9"
14
15 # Retrieve the genus (first word)
16 > genus <- gsub(" .*$", "", organisms)
17
18 > mean_dist_output <- analysis_pca_mean_distance_grouping(pca$x, FACTOR = genus ,
19                                                         xLabs.angle = F, Width = 2, Height = 1.5 ,
20                                                         Filename = "plot_pca_scatter.png")
```



usage example is presented in Code 4.16.

#### 4.3.3.3 *analysis\_genomes\_module\_output*

*analysis\_genomes\_module\_output* takes in the *mcf* matrix (genomes and modules as rows and columns, respectively) and produces a series of analyses and generates a report automatically by default. This function can take in a grouping ‘*FACTOR*’ to split the data corresponding to the genomes (e.g. genus, species, sample, ...) to do taxonomy-level analyses.

This function will:

- 1 report the total number of data sets (genomes) and modules analysed,
- 2 generate a heatmap of the *mcf* of all genomes and modules analysed,
- 3 generate boxplots of the *mcf* across all genomes for each module,
- 4 generate a scatter plot of the standard deviation of the *mcf* across all genomes for each module,
- 5 identify any modules that have a constant (zero-variance) *mcf* across all genomes,
- 6 group the genomes by genus and make a heatmap of the mean *mcf* for each module and genus,
- 7 carry out a PCA analysis, showing the cumulative variance and a PC plot,
- 8 perform a ‘*FACTOR*’-level analysis by visualising the PC plot overlaying the ‘*FACTOR*’ grouping, and measuring the within-group variance (as defined by the ‘*FACTOR*’), using the mean Euclidean distance of the PCs as a proxy for spread.

See the worked out biological example (Section 4.4, Subsection 6) ) and the report generated in this section, available in the GitHub repository.

#### 4.3.4 MetQy visualisation functions

The MetQy *visualisation* family of functions is designed to facilitate the visualisation of the *query* functions as the name suggests. MetQy features five visualisation functions:

- *plot\_heatmap*
- *plot\_scatter\_byFactors*
- *plot\_variance\_boxplot*
- *plot\_scatter*
- *plot\_sunburst*

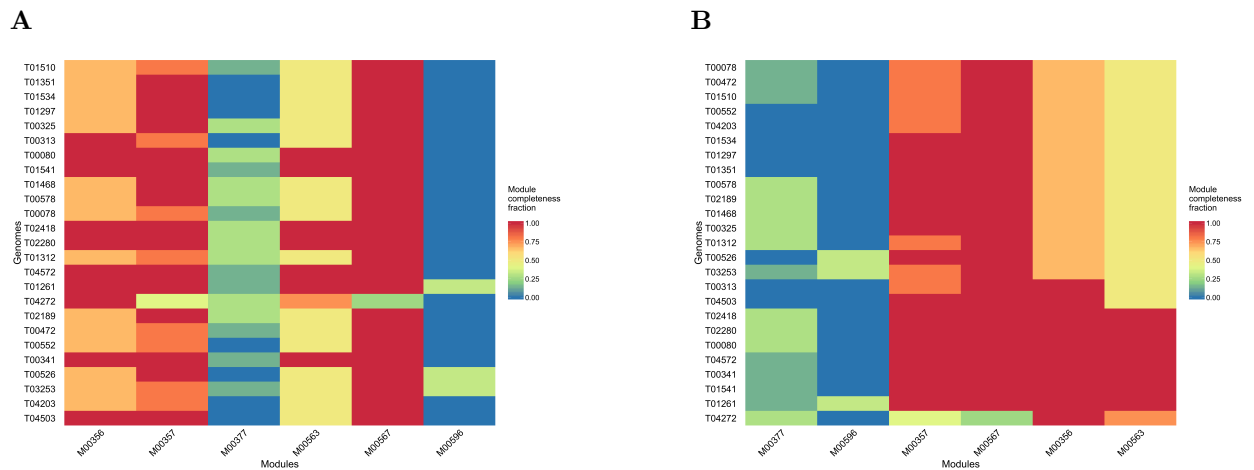
These functions are described next. Note that the function *analysis\_genomes\_module\_output* processes the *query\_genomes\_to\_modules* output (*mcf* matrix) and uses all visualisation functions, except *plot\_sunburst*.

#### 4.3.4.1 *plot\_heatmap*

*plot\_heatmap* is useful to visualise the *mcf* matrix calculated by the *query\_genomes\_to\_modules* function as a colour mapped matrix of the genomes (rows) against modules (columns). Codes 4.17 and 4.18 contain usage examples and the plots produced are shown in Figure 4.1A and 4.1B, respectively.

##### Code 4.17 Usage example for *plot\_heatmap*

```
1 > data(data_example_moduleIDs)
2 > data(data_example_genomeIDs)
3
4 # Calculate the module completion fraction (mcf) for the genomes and modules
  contained in the data objects above.
5 > OUT <- query_genomes_to_modules(data_example_genomeIDs,
6                                   MODULE_ID = data_example_moduleIDs)
7
8 # Make a heatmap of the mcf output from query_genomes_to_modules.
9 > p <- plot_heatmap(OUT$MATRIX, ORDER_MATRIX = F, Filename = "plot_heatmap.png")
```



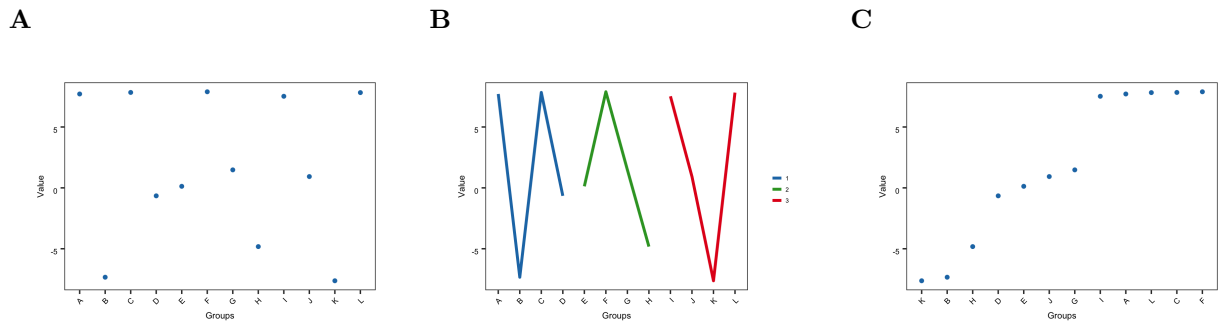
**Figure 4.1** Heatmaps generated by *plot\_heatmap* using Codes 4.17 and 4.18 showing the *mcf* of the genomes provided for the modules specified in the ordered provided (A) and ordered ordered according to the dendrogram resulting from hierarchical clustering (B).

##### Code 4.18 Usage example for *plot\_heatmap* – ordered data

```
1 # Reorder rows and columns according to the dendrogram resulting from hierarchical
  clustering (same data).
2 p <- plot_heatmap(OUT$MATRIX, Filename = "plot_heatmap-ordered.png")
```

#### 4.3.4.2 *plot\_scatter*

This function is useful to visualise a scatter plot made of categorical or string variables associated with numeric values. Codes 4.19, 4.20 and 4.21 show coded examples for 12 groups given randomly generated values and assigned to a ‘*FACTOR*’. This could be equivalent of using organisms’ genus to



**Figure 4.2** Scatter plots generated by *plot\_scatter* using Codes 4.20, 4.21 and 4.21. **A** Plot of the “Values” associated with each group in the order provided. **B** Plot of the “Values” associated with each group grouped by ‘*FACTOR*’. **C** Plot of the “Values” associated with each group in increasing order (manually ordered).

group different genomes, with the mean distance of the *mc*f matrix as the value and the Phylum-level taxonomy membership as the ‘*FACTOR*’. The first example plots the groups in the order given with their respective values. The second example plots the values connected by the ‘*FACTOR*’ membership and the third example shows how the order can be manually adjusted by the user to best represent the data. The plots generated are shown in Figure 4.2.

**Code 4.19** Usage example for *plot\_scatter* – plot group values

```
1 > plot_data_example <- data.frame("Groups" = LETTERS[1:12],
2                                   "Factor" = c(rep(1,4), rep(2,4), rep(3,4)),
3                                   "Value" = runif(12, -10, 10),
4                                   stringsAsFactors = FALSE)
5
6 > p1 <- plot_scatter(plot_data_example, Filename = "plot_scatter.png",
7                     Width = 4, Height = 3)
```

**Code 4.20** Usage example for *plot\_scatter* – plot group values coloured by the ‘*FACTOR*’

```
1 # Separate based on the '\emph{\textsf{FACTOR}}' column
2 > p2 <- plot_scatter(plot_data_example, colBy = 2, Filename="plot_scatter_colBy.png",
3                     Width = 4, Height = 3)
```

**Code 4.21** Usage example for *plot\_scatter* – plot group values in the order given; feature to allow users to better represent data

```
1 # Change plot order to be according to the numeric value
2 > plot_data_example_ordered <- plot_data_example[order(plot_data_example$Value),]
3 > p3 <- plot_scatter(plot_data_example_ordered, Filename="plot_scatter_ordered.png",
4                     Width = 4, Height = 3)
```

#### 4.3.4.3 *plot\_scatter\_byFactors*

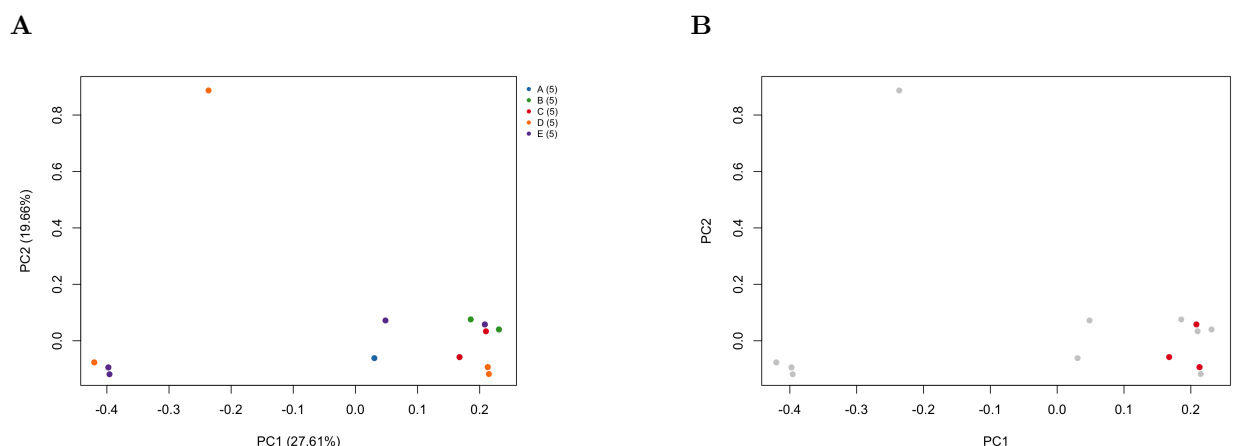
*plot\_scatter\_byFactors* allows the automatic grouping of data as determined by a factor and produces a scatter plot with groups overlaid by colour. This function was designed to be used by *analysis\_pca\_mean\_distance\_grouping* to plot the first two principal components (PCs) resulting from the *mcf* matrix with the data grouped by *factor(s)* (see Section 4.3.3). Code 4.22 and Figure 4.3A show the *mcf* matrix generated using the MetQy example data with artificial labels used as grouping factors (letters A to E). The data corresponding to each group is assigned a different colour. The function supports up to 59 groups. If more are provided, a warning is issued, the plot is not generated and only the number of groups and the number of members per group is returned. This function can also be used to contrast a subset of the data by assigning a group level to the subset and [NA](#). Code 4.23 and Figure 4.3B show an example of only one group being highlighted by colouring those entries while greying the remainder out.

**Code 4.22** Usage example for *plot\_scatter\_byFactors*

```

1 > data(data_example_moduleIDs)
2 > data(data_example_genomeIDs) # length(data_example_genomeIDs) # [1] 25
3
4 # Calculate the module completion fraction (mcf) for the genomes and modules
  contained in the data objects above.
5 > OUT      <- query_genomes_to_modules(data_example_genomeIDs ,
6                                       MODULE_ID = data_example_moduleIDs)
7 # GET PCA
8 > pca <- prcomp(OUT$MATRIX)
9 # Make boxplots of the mcf output from query_genomes_to_modules
10 > this_FACTOR <- rep(LETTERS[1:5],length(data_example_genomeIDs)/5)
11 > plot_output <- plot_scatter_byFactors(pca$x[,1:2],FACTOR = this_FACTOR,
12                                       factor_labs = "random",
13                                       Filename = "plot_scatter_byFactors.png")

```



**Figure 4.3** Scatter plot of the first two principal components (PCs) of the example data shown in Codes 4.22 and 4.23. **A** Data coloured by group. **B** Labels of group C provided, which have been highlighted against the remaining data.

**Code 4.23** Usage example for *plot\_scatter\_byFactors*

```

1 # NAs are omitted, so a single group can be contrasted with overall data
2 this_FACTOR <- c(rep(NA,20),rep(LETTERS[1],5))
3 plot_output <- plot_scatter_byFactors(pca$x[,1:2],FACTOR = this_FACTOR,
4                                     factor_labs = "group-A",
5                                     Filename="plot_scatter_byFactors_single.png")

```

**4.3.4.4** *plot\_sunburst*

KEGG modules have a hierarchical class system, where Class I is the higher level class (e.g. “Pathway module”) and Class III is the most specific (e.g. “Carbohydrate and lipid metabolism”). Sunburst plots enable the representation of hierarchical information by having concentric rings from the higher level at the centre and becoming more specific in an outwards radial fashion. Additionally, numeric information can also be represented by filling corresponding areas with colours according to a colour bar. Therefore, *plot\_sunburst* is useful for the visualisation of the output from *query\_genomes\_to\_modules*. Code 4.24 demonstrates how the example data *data\_example\_sunburst* was generated. Usage examples of *plot\_sunburst* are shown in Codes 4.25, 4.26 and 4.27.

**Code 4.24** Usage example to generate the data for *plot\_sunburst*

```

1 ### GET mcf for some genomes and modules
2 > T_NUMEBERS <- c("T04503","T04203","T03253","T00526","T00341","T00552","T00472",
3                  "T02189","T04272","T01261","T04572","T01312","T02280","T02418",
4                  "T00078","T00578","T01468","T01541","T00080","T00313","T00325",
5                  "T01297","T01534","T01351","T01510")
6
7 > MODULES      <- c("M00087","M00356","M00357","M00563","M00567","M00596","M00530",
8                  "M00377")
9
10 > OUT          <- query_genomes_to_modules(T_NUMEBERS,MODULE_ID = MODULES,
11                                           META.OUT = T, ADD.OUT = T)
12 ### GET THE MODULE NAME AND INFO IN INCREASING ORDER OF SPECIFICITY
13 > names(OUT$METADATA[c(4:6,3)])
14 # [1] "CLASS.I"    "CLASS.II"   "CLASS.III"  "NAME.SHORT"
15
16 > data_example_sunburst <- OUT$METADATA[,c(4:6,3)]
17 # The data contains the name and classes of the 8 modules analysed.

```

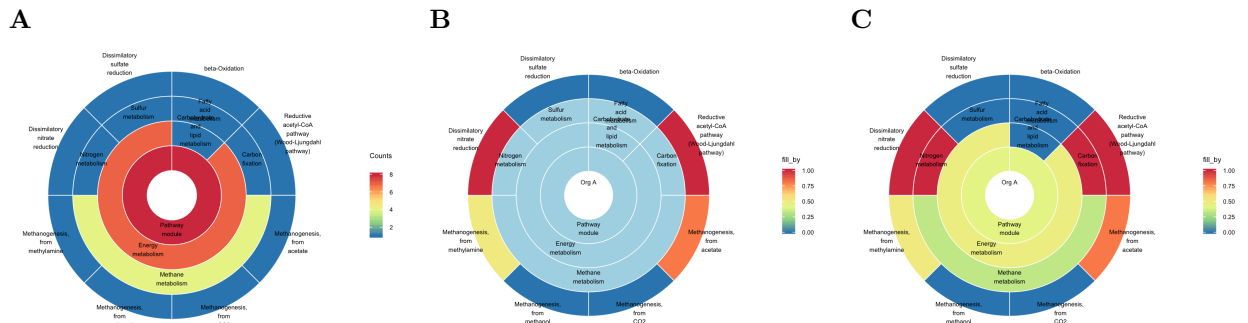
Code 4.25 and Figure 4.4A show how *plot\_sunburst* was used to represent the *mcf* of 8 modules across the genomes analysed. The default colouring is by the “count”, i.e. the number of occurrences for each entry at each level. In this instance *data\_example\_sunburst* contains the name

**Code 4.25** Usage example for *plot\_sunburst*

```

1 > data(data_example_sunburst)
2
3 # Simplest plot using count data (WIDTH scales text size)
4 > plot_data <- plot_sunburst(data_example_sunburst,WIDTH = 8,HEIGHT = 8,
5                             Filename = "plot_sunburst.png")

```



**Figure 4.4** Sunburst plots generated by Codes 4.25, 4.26 and 4.27. **A** Showing the *mcf* of 8 modules across the genomes analysed. The default colouring is by the “count”. **B** The sunburst plot shows *mcf* of a single genome (labelled as ‘Org A’ in the example) in the outer ring, as specified by ‘fill\_by’. **C** Same as (B), with the additional argument ‘fill\_by\_mean = TRUE’ indicating that the inner rings should be coloured by the mean fill\_by value (*mcf*) at each hierarchical level.

and classes of the 8 modules analysed and hence the outer ring is all coloured blue (indicating one as observed in the colour bar).

*plot\_sunburst* can also be used to show the *mcf* of a single genome (labelled as ‘Org A’ in the example; Code 4.26 and Figure 4.4A). In this instance, as ‘fill\_by’ was specified, the colour of the outer ring corresponds to the values provided in `data_example_sunburst.fill_by` (*mcf*). The inner rings have no values and are therefore light blue. Alternatively, the inner rings can be coloured by the mean fill\_by value by indicating ‘fill\_by\_mean = TRUE’ (Code 4.27 and Figure 4.4C).

**Code 4.26** Usage example for *plot\_sunburst* – colouring the outer ring

```
1 > data(data_example_sunburst_fill_by)
2   # data generated by taking the mcf of the first organism
3   # data_example_sunburst_fill_by <- OUT$MATRIX[1,]
4
5 # Specify values to be used for outer ring (lowest level) and change legend name
6   # accordingly
7 > plot_data <- plot_sunburst(data_example_sunburst, centerLabel = "Org A",
8                             fill_by = data_example_sunburst_fill_by,
9                             legend_name = "fill_by", WIDTH = 8, HEIGHT = 8,
10                            Filename = "plot_sunburst_fill_by.pdf")
```

**Code 4.27** Usage example for *plot\_sunburst* – colouring the outer and inner rings

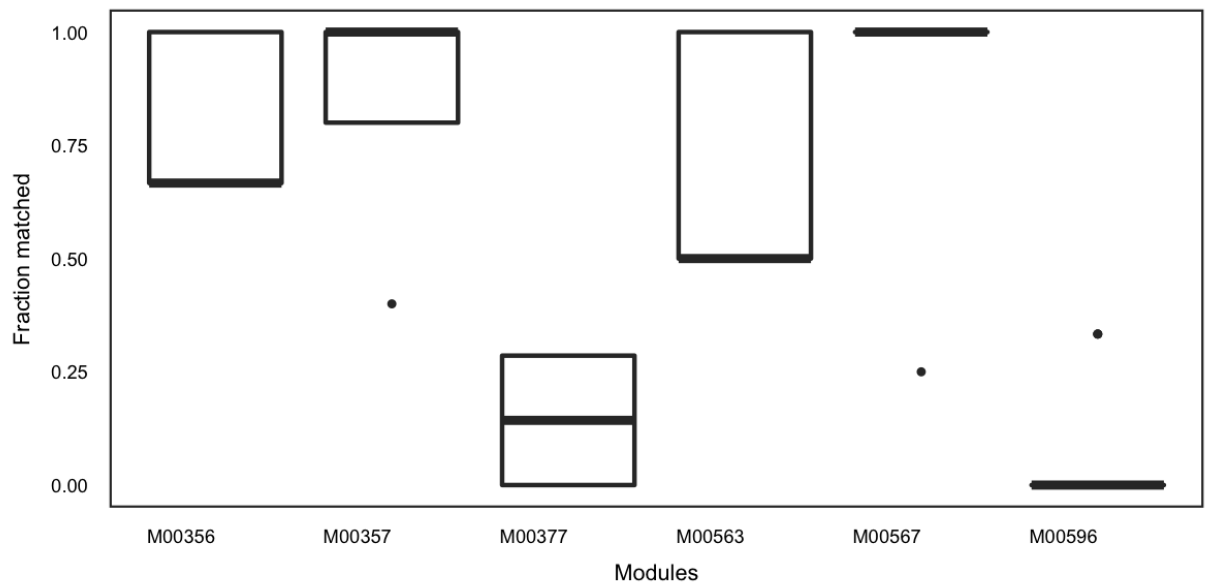
```
1 # Also fill inner rings (levels) according to the mean values determined by fill_by
2 > plot_data <- plot_sunburst(data_example_sunburst, centerLabel = "Org A",
3                             fill_by = data_example_sunburst_fill_by,
4                             fill_by_mean = TRUE, legend_name = "fill_by",
5                             WIDTH = 8, HEIGHT = 8,
6                             Filename = "plot_sunburst_fill_by_mean.pdf")
```

#### 4.3.4.5 *plot\_variance\_boxplot*

*plot\_variance\_boxplot* can be used to summarise the variance of the *mcf* across genomes with a boxplot of the modules analysed. An example can be seen in Code 4.28 and Figure 4.5.

**Code 4.28** Usage example for *plot\_variance\_boxplot*

```
1 # Retrieve example data
2 > data(data_example_moduleIDs)
3 > data(data_example_genomeIDs)
4
5 # Calculate the module completion fraction (mcf) for the genomes and modules
  contained in the data objects above.
6 > OUT      <- query_genomes_to_modules(data_example_genomeIDs ,
7                                       MODULE_ID = data_example_moduleIDs)
8
9 # Make boxplots of the mcf output from query_genomes_to_modules
10 > p <- plot_variance_boxplot(OUT$MATRIX, xLabs_angle = FALSE,
11                             Filename = "plot_variance_boxplot.png",
12                             Width = 4, Height = 2)
```



**Figure 4.5** Boxplot generated by *plot\_variance\_boxplot* using Code 4.28 to summarise the *mcf* variance across genomes for the modules analysed.

## 4.4 Using MetQy: a biological example

This section provides a tutorial-style walkthrough of most of MetQy functions in the context of few basic biological analyses as examples. These are outlined below.

- 1 Use the *query\_genomes\_to\_modules* function to

- 1.1 identify potential methanogens by searching for genomes whose organism name contain “methano”, and

- 1.2 evaluate metabolic processes (module completeness fraction, *mcf*) that are loosely related to anaerobic digestion (AD) processes across these genomes
- 2 Visualise the *mcf* for the AD modules across the selected genomes in KEGG using a heatmap
- 3 Investigate which are the genomes that have a low *mcf* for module ‘M00567’ based on the heatmap
- 4 Use one of these genomes (T04272) and investigate which genes are missing for that module to be complete
- 5 Visualise T04272’s *mcf* for all the modules including module classes using a sunburst plot
- 6 Carry out an automated analysis & report using the *analysis\_genomes\_module\_output* function

This will automatically analyse all the data generated in step 1) and also the data grouped by genus (given as a ‘*FACTOR*’). This function will:

- 6.1 report the total number of data sets (genomes) and modules analysed,
- 6.2 generate a heatmap of the *mcf* of all genomes and modules analysed,
- 6.3 generate boxplots of the *mcf* across all genomes for each module,
- 6.4 generate a scatter plot of the standard deviation of the *mcf* across all genomes for each module,
- 6.5 identify any modules that have a constant (zero-variance) *mcf* across all genomes,
- 6.6 group the genomes by genus and make a heatmap of the mean *mcf* for each module and genus,
- 6.7 carry out a PCA analysis, showing the cumulative variance & a principal component (PC) plot,
- 6.8 perform a genus-level analysis by visualising the PC plot overlaying the genus grouping, and measuring the within-group (genus) variance, using the mean Euclidean distance of the PCs as a proxy for spread.

#### 4.4.1 Identify potential methanogens and evaluate their genomes for selected metabolic processes loosely relating to anaerobic digestion.

We are first interested in evaluating the *mcf* of organisms that have “methano” in the name across loosely AD-related modules, which have been “hand-picked”.

**Code 4.29** Step 1: Identification of organisms and evaluation of metabolic processes

```

1 # Load library
2 > library(MetQy)
3

```



```

4 # Create a folder to store output
5 > output_path <- "Example_1"
6 > dir.create(output_path)
7
8 # Modules loosely related to anaerobic digestion (AD)
9 > AD_modules <- c("M00087", "M00356", "M00357", "M00563", "M00567", "M00596",
10                  "M00530", "M00377")
11
12 # Let the function find the KEGG genomes that have "methano" in the name and
13   calculate the module completeness fraction (mcf) for the AD-relevant modules
14 > query_output <- query_genomes_to_modules(GENOME_INFO = "methano",
15                                           MODULE_ID = AD_modules,
16                                           META_OUT = T, ADD_OUT = T)
17 # Set OUTMODULENAME to TRUE to retrieve back mcf data with module names instead
18   of IDs.
19 # You can also use the module IDs to replace it with the NAME (contained in
20   $METADATA)
21
22 # Retrieve information about matching organisms
23 > nrow(query_output$GENOME_INFO_DATA)
24 # [1] 86 - # of KEGG genomes that matched (partially) the organism name
25
26 > organisms <- query_output$GENOME_INFO_DATA$ORGANISM
27 # There are "Candidate" organisms in the KEGG genome database. We'll remove those.
28 > candidate_index <- grep("candidat", organisms, ignore.case = T)
29 > query_output$GENOME_INFO_DATA[candidate_index,]
30
31 #      ID ORG_ID                                     ORGANISM
32 # 22 T00823    mpl      Candidatus Methanosphaerula palustris E1-9c
33 # 49 T02624    max      Candidatus Methanomethylophilus alvus Mx1201
34 # 51 T02692    mer Candidatus Methanomassiliicoccus intestinalis Issoire-Mx1
35 # 57 T03540    mear      Candidatus Methanoplasma termitum MpT1
36
37 # Remove candidate organisms
38 > organisms <- organisms[-candidate_index]
39 > mcf_matrix <- query_output$MATRIX[-candidate_index,]
40 > genome_info <- query_output$GENOME_INFO_DATA[-candidate_index,]
41 > module_info <- query_output$METADATA[-candidate_index,]
42
43 # Retrieve the GENUS (first word) by removing the rest of the text
44 > genus <- gsub(pattern = "\\s.+$", replacement = "", organisms)
45 > length(unique(genus)) # [1] 27

```

## 4.4.2 Visualise the *mcf* for the user-specified modules across the selected genomes in KEGG using a heatmap

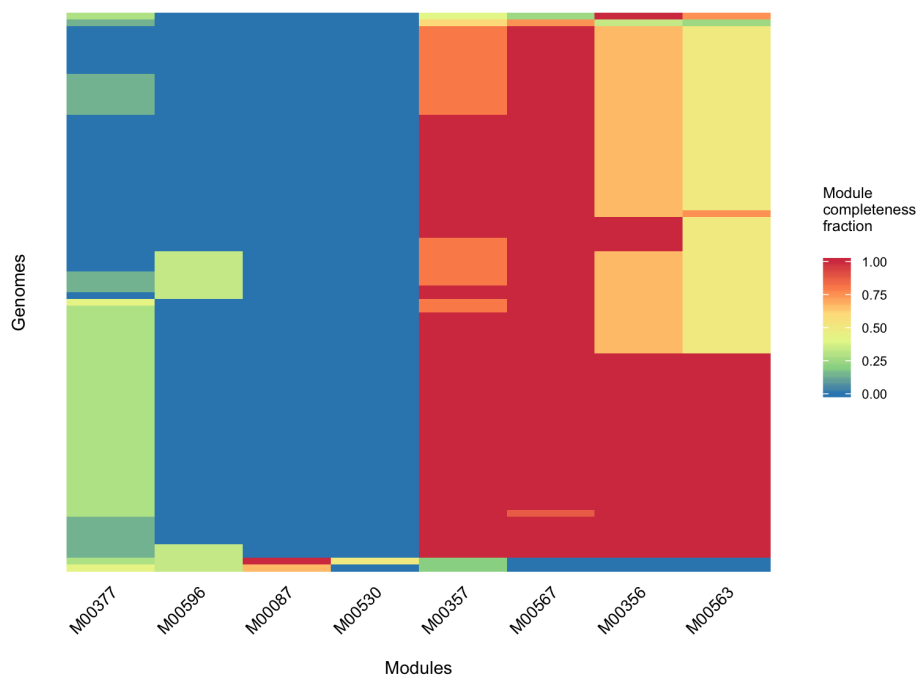
See Code 4.30 and Figure 4.6.

**Code 4.30** Step 2: Visualise *mcfs*

```

1 # Quick look at mcfs
2 > plot_heatmap(mcf_matrix, ORDER_MATRIX = T)
3   # set ORDER_MATRIX to TRUE so that the data is ordered according to
   hierarchical dendrogram
4
5 # The module names are too long making the plot hard to read, so we could manually
   abbreviate them or we can use the module IDs instead (FIGURE NOT SHOWN)
6 > colnames(mcf_matrix)
7 [1] "beta-Oxidation"
8 [2] "Methanogenesis, methanol => methane"
9 [3] "Methanogenesis, acetate => methane"
10 [4] "Reductive acetyl-CoA pathway (Wood-Ljungdahl pathway)"
11 [5] "Dissimilatory nitrate reduction, nitrate => ammonia"
12 [6] "Methanogenesis, methylamine/dimethylamine/trimethylamine => methane"
13 [7] "Methanogenesis, CO2 => methane"
14 [8] "Dissimilatory sulfate reduction, sulfate => H2S"
15
16 > colnames(mcf_matrix) <- query_output$METADATA$MODULE_ID
17
18 # Look at mcfs again
19 > plot_heatmap(mcf_matrix, ORDER_MATRIX = T, Filename = "Example.1/heatmap.png",
20   Height = 4, Width = 5.4)
21   # set ORDER_MATRIX to TRUE so that the data is ordered according to
   hierarchical dendrogram

```



**Figure 4.6** Step 2: Heatmap of the *mcf* of the organisms with “methano” in their name and the loosely AD-related modules

### 4.4.3 Investigate which are the genomes that have a low *mcf* for module ‘M00567’ based on the heatmap

The heatmap shows that module ‘M00567’ (Methanogenesis, CO<sub>2</sub> => methane) has a widespread “good” coverage – *mcf* >= 0.75 in 96% of genomes. We can investigate which genomes those are, as well as get more information about the module (Code 4.31).

**Code 4.31** Step 3: investigating module ‘M00567’

```

1 > names(module_info)
2 [1] "MODULE_ID" "MODULENAME" "NAME_SHORT" "CLASS_I" "CLASS_II" "CLASS_III"
3 [7] "DEFINITION" "OPTIONAL"
4
5 > module_info[which(module_info$MODULE_ID=="M00567"),1:3]
6   MODULE_ID      MODULENAME      NAME_SHORT
7   M00567      Methanogenesis , CO2 => methane      Methanogenesis , CO2 => methane
8
9 > module_info[which(module_info$MODULE_ID=="M00567"),4:6]
10   CLASS_I      CLASS_II      CLASS_III
11   Pathway module      Energy metabolism      Methane metabolism
12
13 > small_M00567_index <- which(mcf_matrix[,7] < 0.75)
14 > mcf_matrix[small_M00567_index,7]
15 T03209 T03230 T04272
16   0.00   0.00   0.25 # mcf
17
18 # Look at the corresponding genomes (conserved order)
19 > genome_info[small_M00567_index,]
20   ID   ORG.ID      ORGANISM
21   T03209   bmet      Bacillus methanolicus MGA3
22   T03230   amq      Amycolatopsis methanolica 239
23   T04272   marc      Methanogenic archaeon ISO4-H5

```

### 4.4.4 Find out which genes are missing for module ‘M00567’ to be complete for genome ‘T04272’

Investigate which genes are missing for ‘M00567’ to be complete in genome ‘T04272’ (*Methanogenic archaeon* ISO4-H5). ‘T04272’ has an *mcf* for module ‘M00567’ of 0.25. This means that only a quarter of the blocks needed for the module to be complete are present (Code 4.32). As expected, only 2 out of 8 module definition blocks are present (the bottom two). ‘T04272’ has not been annotated with any of The K numbers flanked by asterisks (\*). Note that ‘T04272’ doesn’t need to have *ALL* the genes listed on the ‘MISSING\_GENES’ column; it only needs those to have a complete block according to the block definition (‘BLOCK\_DEF’, logical expression). For instance, in order to have a complete first block, ‘T04272’ requires the following KEGG orthologs:

ALL THESE: ‘K00200’, ‘K00201’, ‘K00202’ and ‘K00203’

AND ONE OF THESE: ‘K00205’, ‘K11260’ or ‘K00204’.

**Code 4.32** Step 4: investigating genes missing from genome ‘T04272’

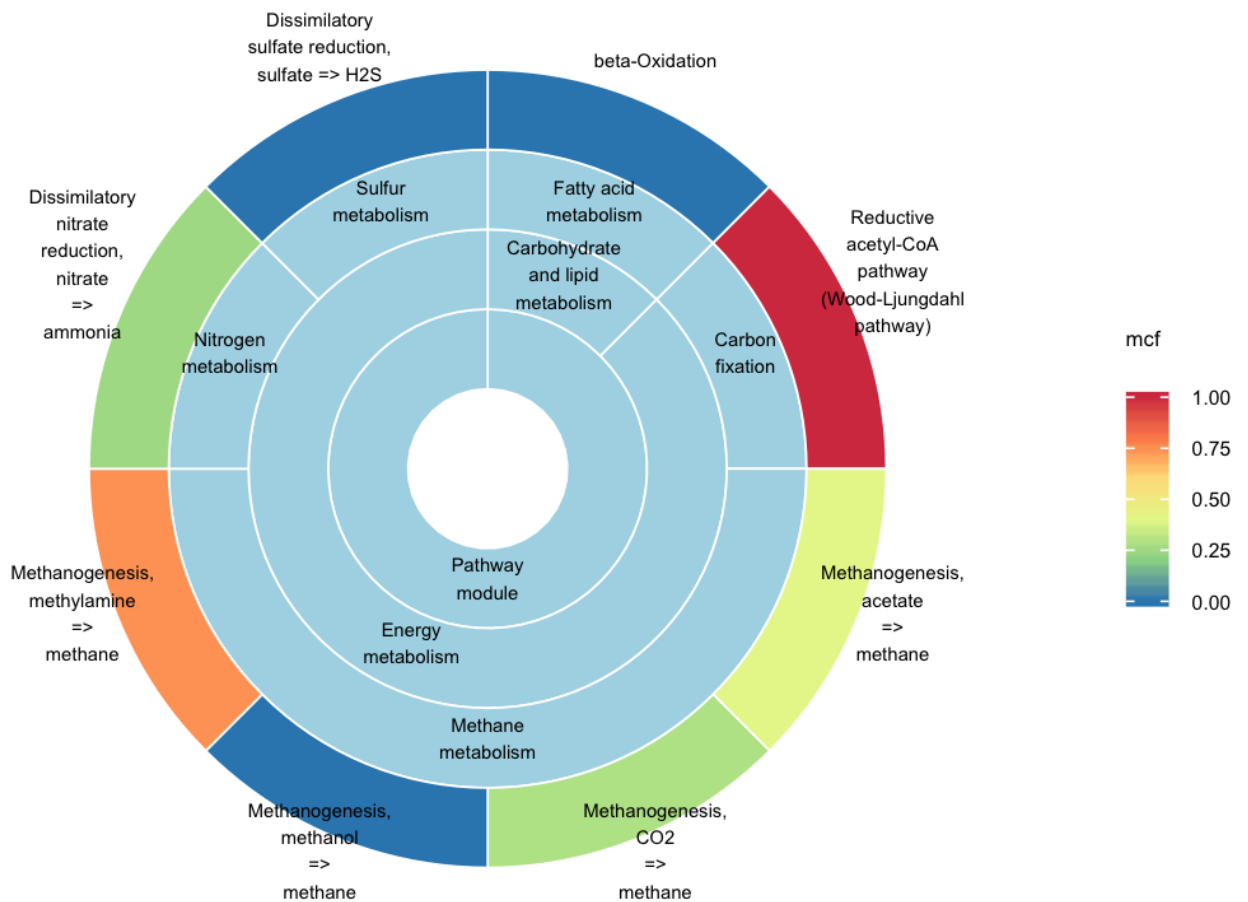
```

1 ## Retrieve the missing K numbers required for module "M00567"
2 > T04272_M00567_missing <- query_missingGenes_from_module(GENOME = "T04272",
3                                                             MODULE_ID = "M00567",
4                                                             PRINT_TO_SCREEN = F)
5
6 > print(T04272_M00567_missing)
7
8      block_No PRESENT BLOCK_DEF
9      1       1      0 *K00200*&*K00201*&*K00202*&*K00203*&(*K00205*|*K11260*|*K00204*)
10     2       2      0 *K00672*
11     3       3      0 *K01499*
12     4       4      0 *K00319*|*K13942*
13     5       5      0 *K00320*
14     6       6      0 *K00577*&*K00578*&*K00579*&*K00580*&*K00581*&*K00584*
15     7       7      1 K00399&K00401&K00402
16     8       8      1 K03388&K03389&K03390
17
18      MISSING_GENES
19      1 K00200;K00201;K00202;K00203;K00205;K11260;K00204
20      2 K00672
21      3 K01499
22      4 K00319;K13942
23      5 K00320
24      6 K00577;K00578;K00579;K00580;K00581;K00584
25      7
26      8

```

#### 4.4.5 Use a sunburst plot to visualise ‘T04272’s *mcf* for all the modules including module information.

We can also look more closely to ‘T04272’s *mcf* with module information to have more of a context and potentially gain more insight.



**Figure 4.7** Step 5: Heatmap of the *mcf* of the organisms with “methano” in their name and the loosely AD-related modules

**Code 4.33** Step 5: visualising ‘T04272’s *mcf* using a sunburst plot

```

1 ## RETREIVE GENOME "T04272"'s mcf
2 > T04272_mcf <- mcf.matrix[which(rownames(mcf.matrix)=="T04272"),]
3
4 ## MODULE INFORMATION
5 > AD_module_info <- query_output$METADATA[c(4:6,3)]
6 > names(AD_module_info)
7 # [1] "CLASS_I"      "CLASS_II"     "CLASS_III"    "NAME_SHORT"
8
9 ## Manually edit text for nicer output
10 > AD_module_info[6,4] <- "Methanogenesis, methylamine => methane"
11
12 ## Replace spaces with new lines to make labels neater
13 > for(c in 1:ncol(AD_module_info))
14   AD_module_info[,c] <- gsub(" ", "\n", AD_module_info[,c])
15
16 ## Manually edit text for nicer output
17 > AD_module_info[1,2] <- "Carbohydrate\nand lipid\nmetabolism"
18 > AD_module_info[1,3] <- "Fatty acid\nmetabolism"
19 > AD_module_info[8,4] <- "Dissimilatory\nsulfate reduction,\nsulfate => H2S"
20
21 ## DISPLAY PLOT
22 > sunburst_output <- plot_sunburst(AD_module_info, fill_by = T04272_mcf)
23
24 ## SAVE PLOT TO FILE
25 > sunburst_output <- plot_sunburst(AD_module_info, fill_by = T04272_mcf,
26                                   legend_name = "mcf\n", WIDTH = 4, HEIGHT = 4,
27                                   Filename = "Example_1/T04272_sunburst.png")

```

#### 4.4.6 Carry out an automated analysis

We can use the *analysis\_genomes\_module\_output* function to analyse all the data generated in step 1 and also the data grouped by genus (given as a ‘*FACTOR*’), generating a report in the process.

This functions allows some analyses to be made based on grouping the data as defined by ‘*FACTOR*’(s). Thus by defining ‘*FACTOR*’ (which can be one factor or multiple, if it’s a list), the mean *mcf* and the standard deviation of the *mcf* are visualised as a heatmap with a row for every group. Here we provide the genus as a grouping ‘*FACTOR*’. Furthermore, the function generates a PC plot with data colour-coded according to the grouping. Finally, the function calculates the mean Euclidean distances from the PC space for each group as a proxy measurement for within-group variation (only for groups with more than one member). As mentioned at the beginning of this section, the report will:

- 1 report the total number of data sets (genomes) and modules analysed,
- 2 generate a heatmap of the *mcf* of all genomes and modules analysed (Figure 4.8),
- 3 generate boxplots of the *mcf* across all genomes for each module (Figure 4.9),
- 4 generate a scatter plot of the standard deviation of the *mcf* across all genomes for each module (Figure 4.10),
- 5 identify any modules that have a constant (zero-variance) *mcf* across all genomes,

- 6 group the genomes by genus and make a heatmap of the mean and the standard deviation of the *mcf* for each module and genus (Figures 4.11 and 4.12),
- 7 carry out a PCA analysis, showing the cumulative variance & a principal component (PC) plot (Figure 4.13),
- 8 perform a genus-level analysis by visualising the PC plot overlaying the genus grouping, and measuring the within-group (genus) variance, using the mean Euclidean distance of the PCs as a proxy for spread (Figures 4.14 and 4.15).

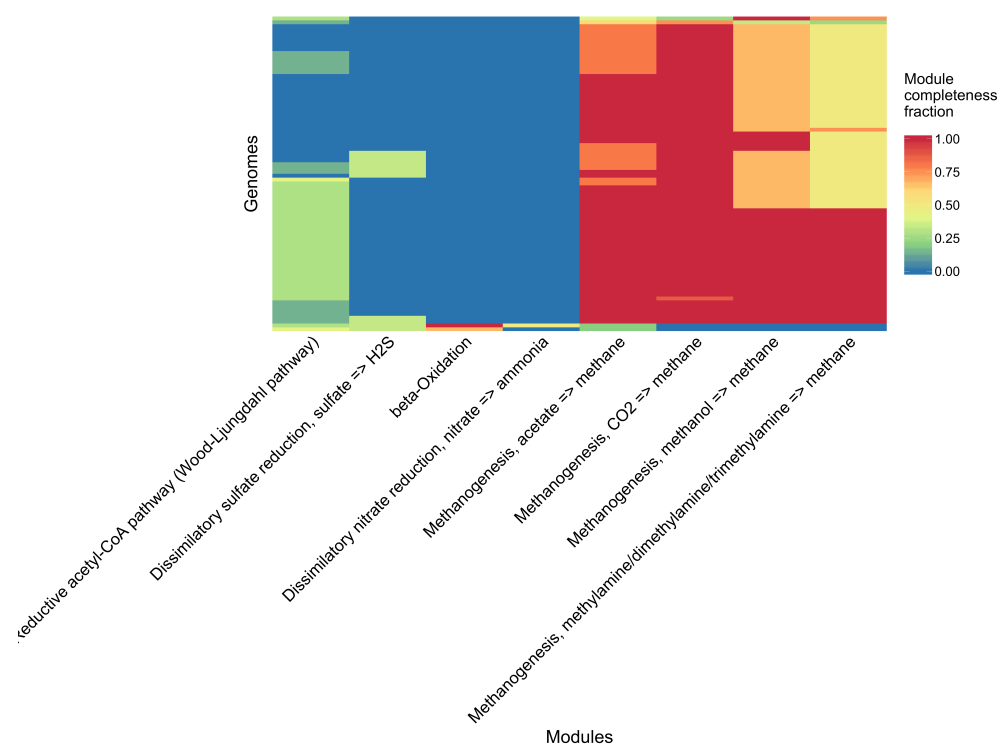
The figures included in the report are shown below by section. The report generated by this function as shown in Code 4.34 is available in the GitHub repository. More details are available in the function documentation (Section E.2).

**Code 4.34** Step 6: automated analysis

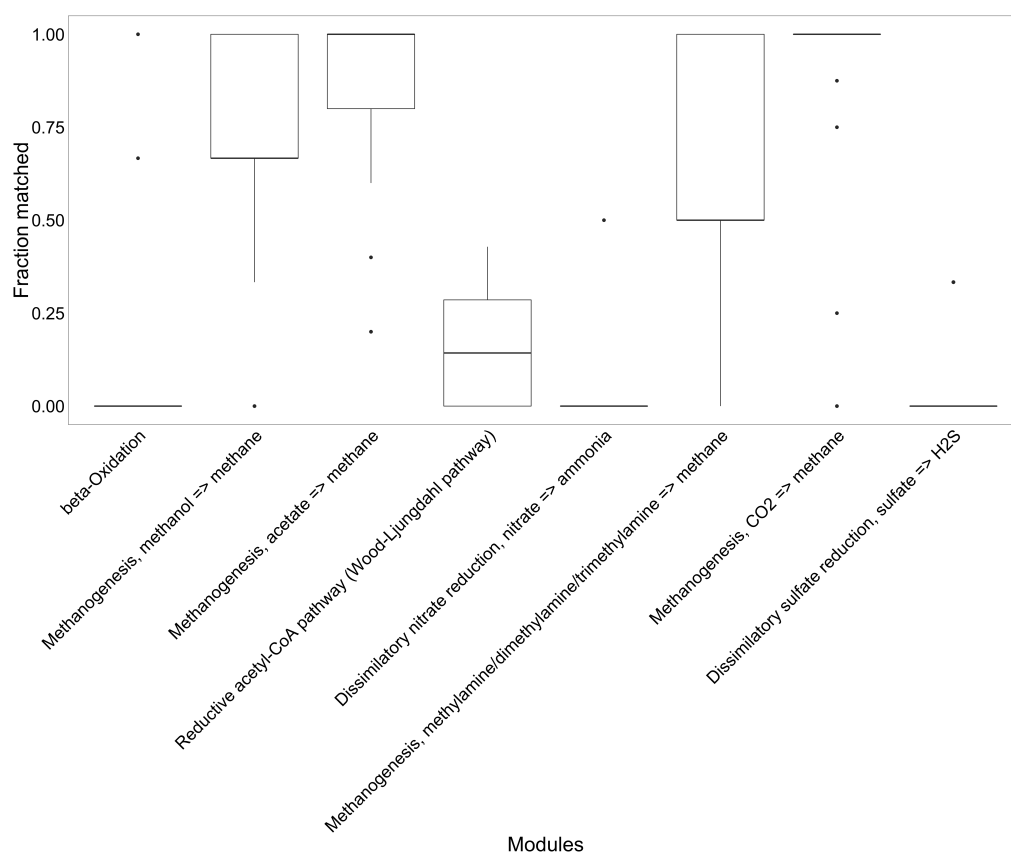
```

1 ## Use the analysis function to process the output from 'query-genomes-to-modules'
  and generate a LaTeX report
2 # Use the genus names we have previously generated to group our data.
3 > useFactor <- list("genusFactor" = genus)
4   # By naming the list item ("genusFactor"), we control the naming convention for
  the files generated. It would be FACTOR by default.
5
6 > analysis_output <- analysis_genomes_module_output(FRACTIONMATRIX = mcf_matrix,
7                                                     outputPath = output_path,
8                                                     figType = ".png",
9                                                     FACTOR = useFactor)
10
11 # GENERATES THE FOLLOWING FILES IN THE OUTPUT FOLDER:
12 #   PCA/
13 #     module_mean_dist_output_FACTOR.rda
14 #     pca_plot.png
15 #     pca_var.png
16 #     pca.rda
17 #     plot_mean_dist_genusFactor.png
18 #     plot_scatter_genusFactor.png
19 #     module_allOrgs_sd_boxplot.png
20 #     module_allOrgs_sd.png
21 #     module_allOrgs.png
22 #     module_constant_presence.txt – information about modules that might be
  absent, present or with the same value across all the genomes analysed for a
  specific module.
23 #     module_mean_genusFactor.png
24 #     module_sd_genusFactor.png
25 #     module_output_plots.rda
26 #     report.tex
27
28 # The files with 'genusFactor' in the name were generated from using the 'FACTOR'
  argument, as we labelled the list in the definition.

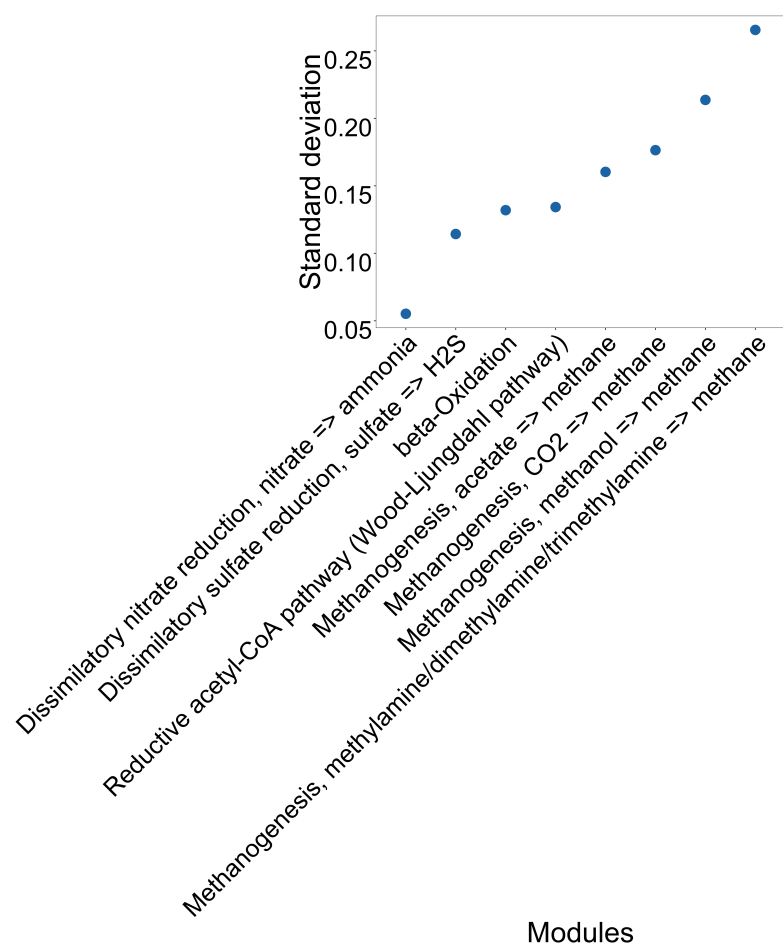
```



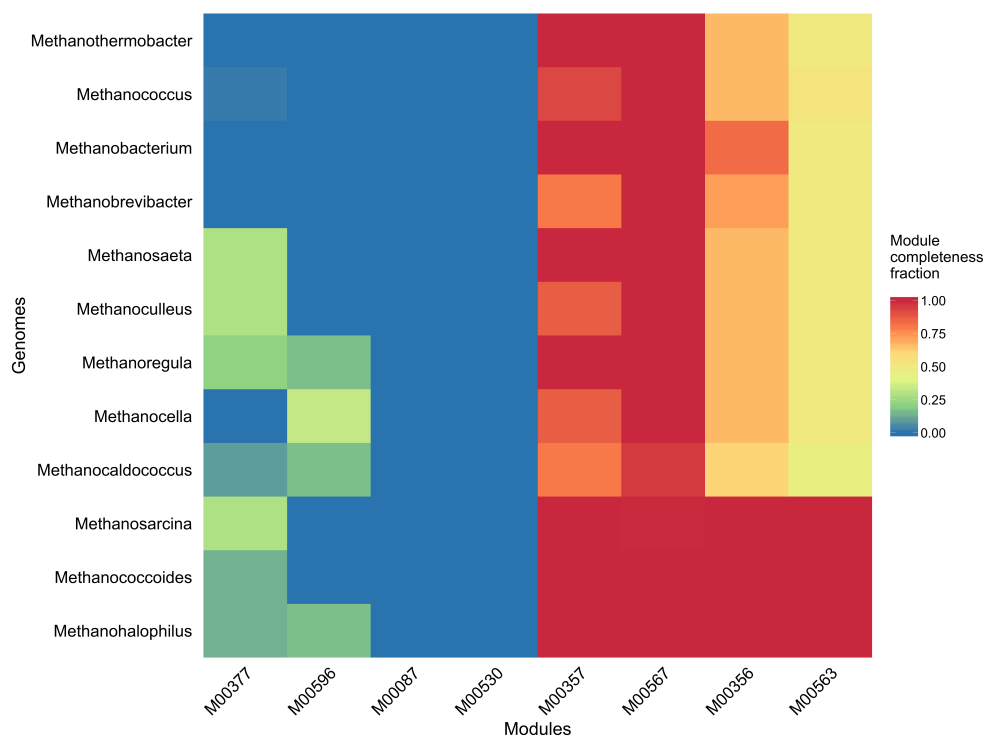
**Figure 4.8** Step 6.2: Heatmap of the *mcf* grouped by genus for the modules specified (the same as Figure 4.6).



**Figure 4.9** Step 6.3: Boxplot of the *mcf* for every module.

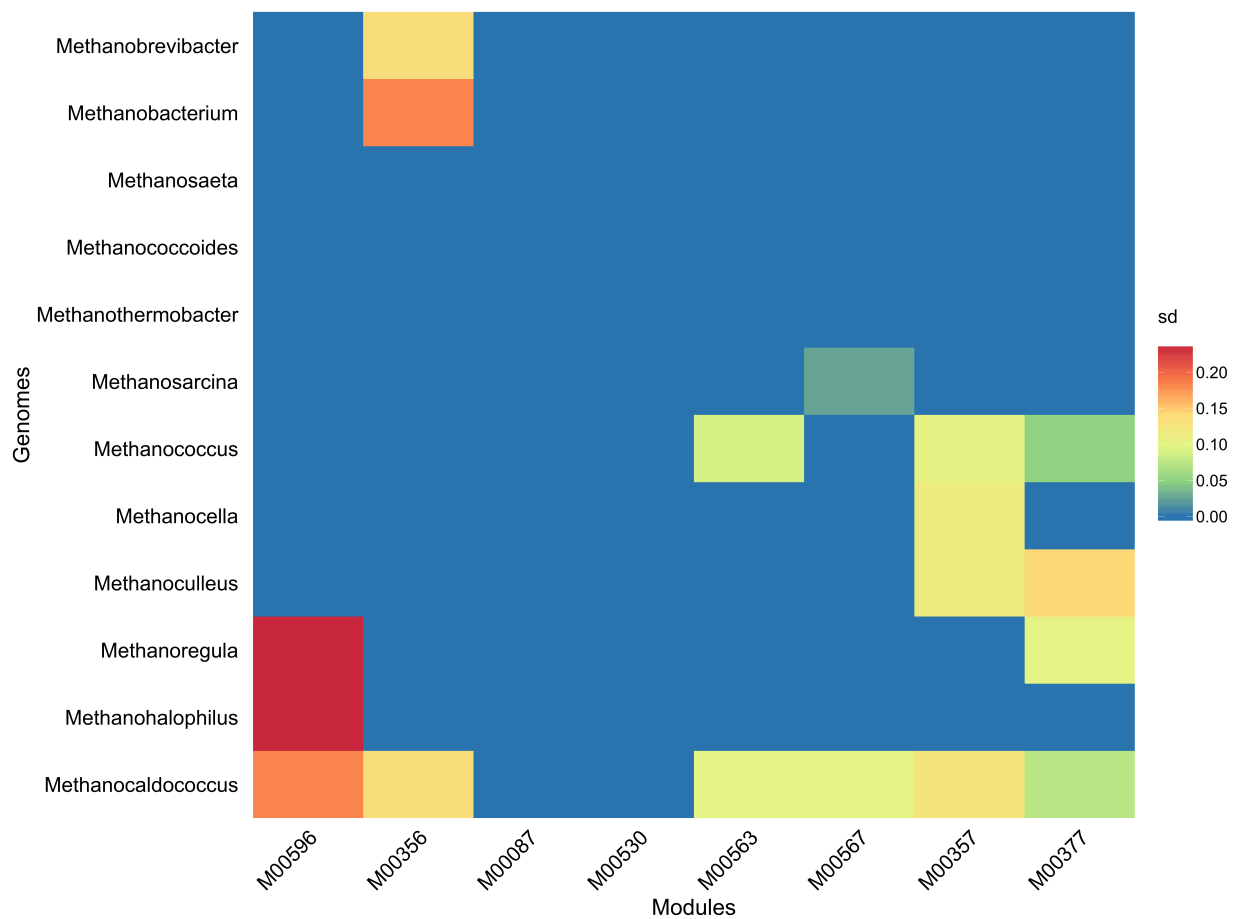


**Figure 4.10** Step 6.4: Scatter plot of the standard deviation of the *mcf* for every module.

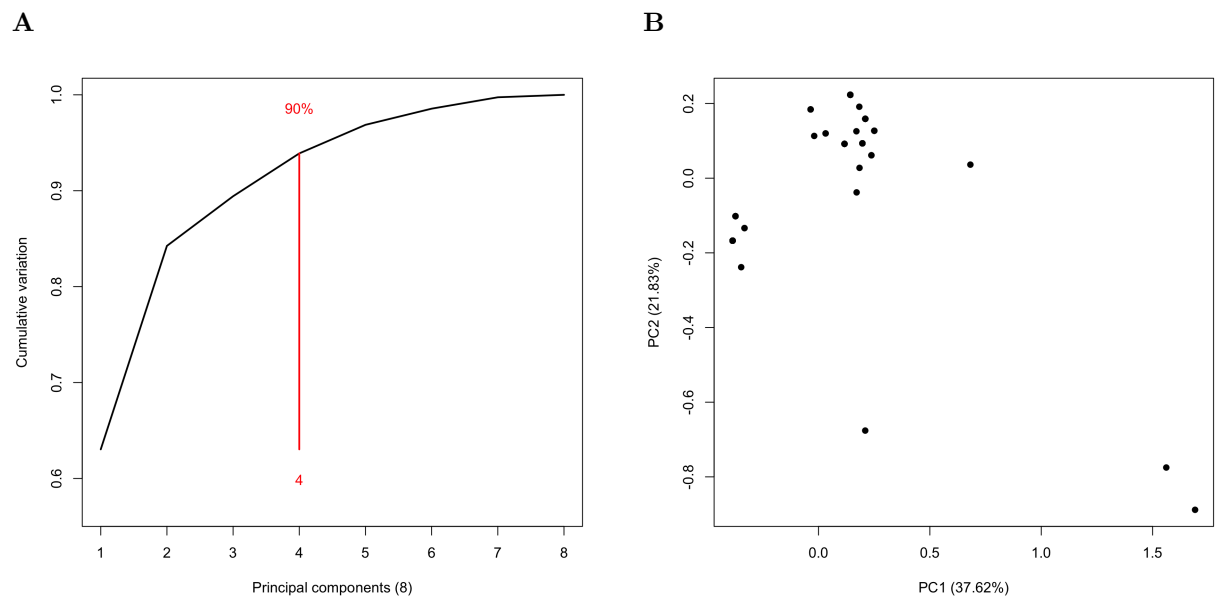


**Figure 4.11** Step 6.6: Heatmap of the mean *mcf* grouped by genus for the modules specified.

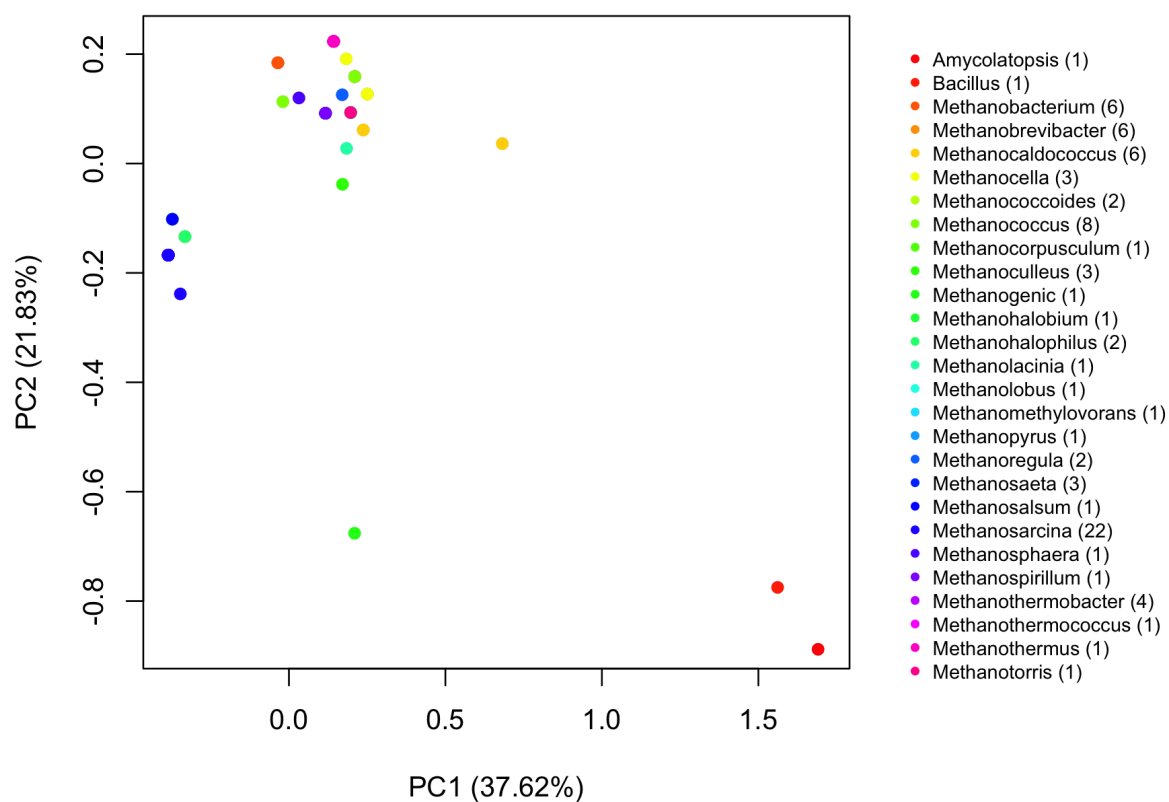




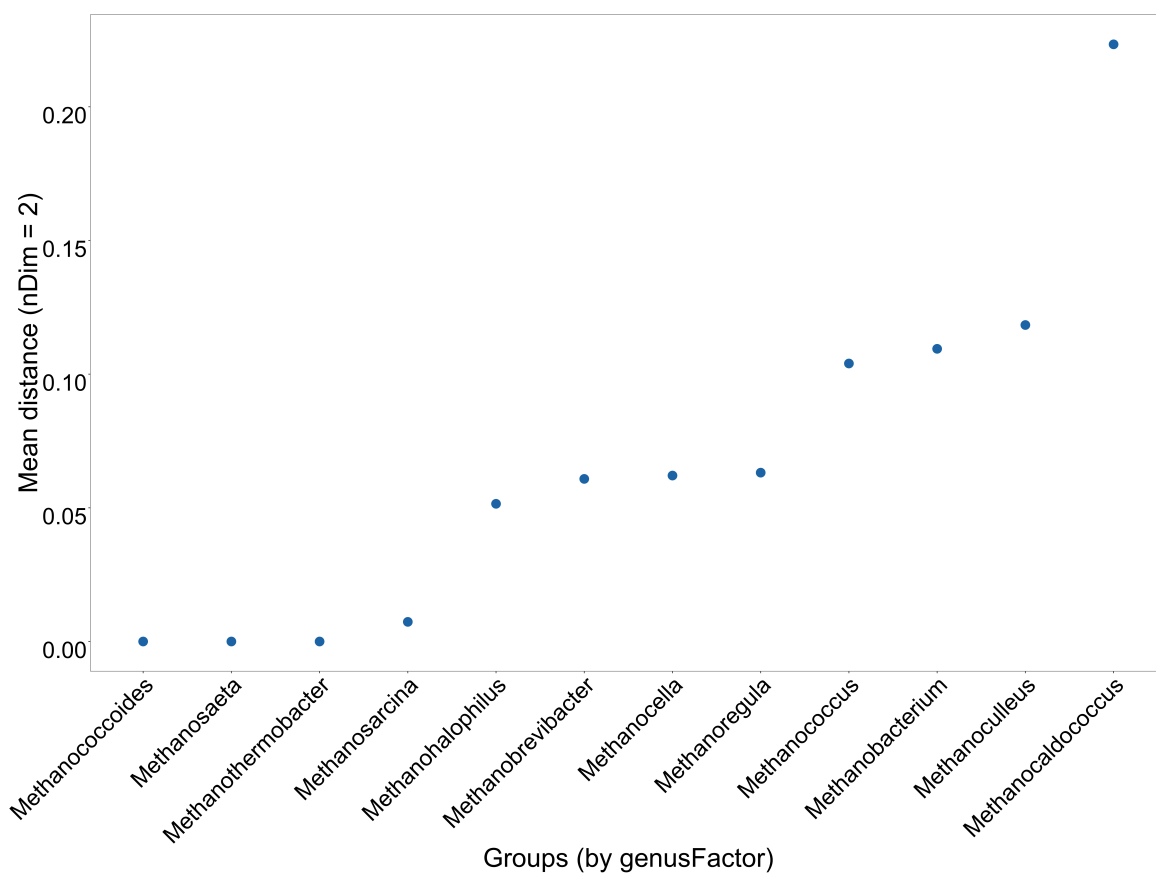
**Figure 4.12** Step 6.6: Heatmap of the standard deviation of the *mcf* grouped by genus for the modules specified.



**Figure 4.13** Step 6.7: PCA analysis of all the data. **A** Cumulative variance captured by the PCs. It can be observed that 4 PCs are required to capture at least 90% of the variance. This is higher than would be expected if the data were highly correlated. **B** Scatter plot of the first two principal components (PCs).



**Figure 4.14** Steps 6.8: Scatter plot of the first two principal components (PCs) colour coded by genus.



**Figure 4.15** Steps 6.8: Mean Euclidean distance calculated from the PCA of the *mcf* for genus groups with more than one member.

## 4.5 Discussion

The aim of the work presented in this chapter was to develop a computational tool, **MetQy**, to enable the data mining and downstream analyses of the relationship between genomic data and biological function (as defined by KEGG modules) contained within the KEGG databases. An overview of the four main families of functions of the R package **MetQy**, namely *parsing*, *query*, *analysis* and *visualisation*, was presented. A useful feature of **MetQy** is its visualisation functionality. Most of these functions make use of **ggplot2**, a graphics package that is very powerful but requires considerable knowledge to use. **MetQy** bridges this gap by providing plotting functions that use advanced R and **ggplot2** knowledge, but are easy to use. For instance, *plot\_scatter* facilitates the generation of a scatter plot with a nominal variable on the x-axis, while *plot\_sunburst* visualises numerical data associated with hierarchical information in the form of a sunburst plot, a non-trivial task. Most, if not all, of the visualisation functions could be valuable for alternative applications, as they are commonly used for data visualisation and have been designed for ease-of-use and provide some flexibility. Additionally, the source code of these visualisation functions could be used as starting points for users with limited **ggplot2** experience.

Biological functions, or traits, have been recorded in multiple databases or in literature-based compilations of functional assignment, or a combination of both (Farrell et al., 2018), e.g. BioCyc and KEGG (Caspi et al., 2012; Kanehisa et al., 2017). Databases have been populated based on specific genes associated with specific functions or metabolic pathways, such as KEGG (Kanehisa et al., 2017), BioCyc (Caspi et al., 2012) and Clusters of Orthologous Groups of proteins (COGs) database<sup>10</sup> (Tatusov et al., 2000). On the other hand, literature-based functional assignments have been compiled for a range of organisms (Louca et al., 2016, 2017) or specific ones, such as methanogenesis (Jablonski et al., 2015), which tend to be less pathway-centric.

The number of functions or traits considered vary across database and compilation. The bulk of the classifications in the Functional Annotation of Prokaryotic Taxa (FAPROTAX) database comes from Bergey’s Manual of Systematic Bacteriology (Whitman et al., 2015), and it currently contains 84 phenotypic traits associated with 4600 microbial taxonomic groups (Louca et al., 2016), while COG has 26 functional categories (Galperin et al., 2015). On the other hand, there are 780 KEGG modules (as of 20/02/2018), suggesting that a greater range of functions and traits could be explored.

In this context and that of the work presented in this thesis, the most relevant *query* function within **MetQy** is *query\_genomes\_to\_modules*, as it allows the evaluation of genomes and genes within KEGG for metabolic functionality. This function embodies an improvement over KEGG’s tool, KEGG Mapper (Kanehisa et al., 2012), as the user can define a set of modules to be

---

<sup>10</sup><http://www.ncbi.nlm.nih.gov/COG/>

evaluated. Furthermore, the introduction of the *module completeness fraction* (*mcf*) informs the user on the coverage of all the selected modules, rather than only apprising them of the subset of KEGG modules with one or no blocks missing. Moreover, *query\_genomes\_to\_modules* was designed for bulk analyses to be carried out, while KEGG Mapper relies on manually-entered searches. Finally, performing the *mcf* evaluation in the R environment facilitates downstream processing, analysis and visualisation.

An advantage of **MetQy** is that tutorial-like examples with example data have been developed to maximise the package’s usability, in addition to the detailed information available as part of the standard R documentation. As an R package, **MetQy** is open-source, enabling the extension of its functionality beyond questions presented in this work. Some *query* functions, for example *query\_genes\_to\_genomes*, were developed based on comments made by participants at events where the author presented this work in the form of a poster of what they would consider “useful” functions<sup>11 12</sup>. However, the extensions could be carried out by any user, and as of 08/08/2018, the **MetQy** GitHub repository has been forked twice (copied to another repository). This suggests that there is a community interested in either proposing changes or in using it as a starting point for other ideas.

**MetQy** facilitates the general usability of the KEGG databases and allows users to gain qualitative information about the functional capacity of a given genome or gene set. Anticipated uses of the tool range across disciplines. For synthetic biologists, it could facilitate the design and guiding of metabolic engineering studies by identifying missing genes needed for an organism to have a complete KEGG module, and identifying KEGG genomes with desired metabolic capabilities. For systems biology applications, it allows identification of key physiological features of organisms and development of stoichiometric metabolic models by analysing module completeness in specific genomes and identifying transporter modules and carbon utilisation routes in genomes. Finally, in microbial ecology, **MetQy** can allow species-function mappings in metagenomes, and insights into functional capabilities of ecological groups by analysing the metabolic capacity of novel genomes from metagenomic studies. Organisms can be put into different functional groups, and the functional profiles of different environments compared.

**MetQy** could also be extended to include additional KEGG databases, such as KEGG pathway. Additionally, it could be incorporated and further developed as part of a pipeline within the R environment, using other KEGG-related packages. For instance, non-FTP users could take advantage of *KEGGREST*<sup>13</sup> package (Tenenbaum, 2018), which implements the REST-style

<sup>11</sup>School of Life Science Postgraduate Symposium (March 2017), University of Warwick, Uk

<sup>12</sup>Joint ICGEB-ICTP-APCTP Workshop on Systems Biology and Molecular Economy of Microbial Communities (July 2017), ICTP, IT

<sup>13</sup><https://bioconductor.org/packages/release/bioc/vignettes/KEGGREST/inst/doc/KEGGREST-vignette.html>

KEGG Application Programming Interface (KEGG API<sup>14</sup>) to extract KEGG data from the FTP resources. Graph analysis could be implemented using the *KEGGGraph* package (Zhang and Wiemann, 2009). *Pathview* (Luo and Brouwer, 2013) could be used for pathway-based data integration and visualization. *KEGGprofile* (Zhao et al., 2017a) could be used to integrate the expression profiles and the function annotation in KEGG pathway maps. Finally, *clusterProfiler* (Yu et al., 2012) could be used to implement statistical analyses of functional profiles for genes and gene clusters.

A remaining challenge is the incorporation of thermodynamic data to this workflow. Many computational approaches have used thermodynamic information to constrain models, such as Beard (2004); De Martino et al. (2014); Henry et al. (2006, 2007); Müller and Bockmayr (2013) and Wodke et al. (2014). However, these constraints are limited to the directionality of the reactions. This task is challenging for several reasons, namely data availability, standardisation of the information, especially regarding reaction notation and identification, and the impact of the environmental conditions on the thermodynamics. Seemingly, there are three literature-based compilations, namely those by Thauer et al. (1977), Alberty (2001) and from The National Institute of Standards and Technology (NIST). Although these are extensive, the first two are text-based and require digitisation. NIST contains a database of “Thermodynamics of Enzyme-Catalyzed Reactions”<sup>15</sup>. While the database is manually searchable through a web-interface<sup>16</sup>, NIST does not provide the complete database, either through a downloadable file or FTP access. This would require laborious attempts to collect all the relevant information. In any case, manual curation would be required to establish the relationship between the reactions described in these databases with the identifiers of reference databases, such as KEGG.

Having sufficient information to implement the use of redox values for metabolic half-reactions, as discussed in Section 1.2 and represented in Figures 1.2 and 1.3, would open up the automated analyses of metabolic functions in the context of biogeochemistry processes, such as  $\text{NO}_3^-/\text{NO}_2^-$  respiration or methanogenesis (Falkowski et al., 2008). Finally, even if the existing thermodynamic data were digitised and incorporated into useful databases, it is worth keeping in mind that environmental conditions, such as compound concentrations, temperature and pH, have an impact on the thermodynamics of any reaction (see Section 1.2, particularly Equation 1.2). Therefore, there would be limitations to the usability of the data unless more sophisticated modelling and data analytic approaches were developed.

<sup>14</sup><https://www.kegg.jp/kegg/docs/keggapi.html>

<sup>15</sup><https://randr.nist.gov/enzyme/Default.aspx>

<sup>16</sup>e.g. search for EC 1.1.1.1 and ID36EUL/ADL\_7:  
[https://randr.nist.gov/enzyme/DataDetails.aspx?ID=36EUL/ADL\\_7&finalterm=1.1.1.1&data=enzyme](https://randr.nist.gov/enzyme/DataDetails.aspx?ID=36EUL/ADL_7&finalterm=1.1.1.1&data=enzyme)

## Chapter 5

# Identification of anodic and cathodic organisms

### *Towards testing the “syntrophy over wires” hypothesis*

#### 5.1 Introduction

The work presented in this thesis has so far been divided into experimental work carried out to investigate the “syntrophy over wires” hypothesis and the development of a computational tool, **MetQy**, to enable the investigation of the relationship between genes and genomes and biological function. This chapter aims to bridge the gap by demonstrating how **MetQy** can be used to inform experiments where other organisms could potentially be paired as syntrophic partners over wires. The organisms that use the electrode as an electron sink, substituting Dv, are referred to as anodic, while those that use the electrode as an electron source, substituting Mm, are referred to as cathodic.

In order to identify the alternative anodic and cathodic organisms, the potential mechanisms by which Dv and Mm carry out extracellular electron transfer (EET) and interact with the electrodes (presented in Chapter 3, Figure 3.1 and Tables 3.1 and 3.2) were used. The K numbers corresponding to the genes encoding for the relevant enzymes and proteins were manually identified by using the KEGG website search engine ([www.kegg.jp](http://www.kegg.jp)). These K numbers were used as input to the **MetQy** function `query_genes_to_genomes` to retrieve the KEGG genomes that were annotated with the K numbers. The code used to implement this approach can be found in Section 5.4.1, Code 5.1.

## 5.2 Results

The potential mechanisms by which Dv and Mm interact with electrodes were summarised in the introduction to Chapter 3. The genes for the proteins involved in the proposed mechanisms were manually mapped to K numbers by searching for the gene ID (RefSeq for Dv and GenBank for Mm) using the KEGG website. When a K number was assigned to an Enzyme Commission (EC) number, this was also noted.

The available information in the literature suggested that Dv could perform indirect and direct long-range electron transfer. The former could be achieved by the use of  $H_2$ , produced by the hydrogenases *Coo*, *Hmc*, *Hyn* and *Hyd*, as an inorganic electron carrier (Walker et al., 2009) and organic electron carriers such as c-type cytochromes (biosynthesised by *Ccm*) and *Fdh* (Croese et al., 2011). The latter is arguably achieved by the presence of flagellar proteins (*Flg*) identified by Walker et al. (2009), as proposed by Croese et al. (2011). Table 5.1 shows that only two of the eight *coo* genes have been mapped to a K number. None of the *hmc* genes has been mapped to a K number, leading to the exclusion of this protein from the analysis. Every *hyd* and *hyn* genes was mapped to a different K and EC number. All 14 *flg* genes were mapped to K numbers. The four *ccm* genes were mapped to K numbers. However, there was a discrepancy between the K number and the RefSeq gene for two of them, as K02194 and K02195 refer to heme transport proteins instead of c-type cytochrome related proteins. Therefore, these two K numbers were excluded from the analysis. Finally, two of the three *fdh* genes were mapped to the same K number, effectively contributing two K numbers to the analysis.

The literature suggested that Mm could perform indirect electron transfer through the use of  $H_2$ , produced by the hydrogenases *Fru*, *Frc*, *Hmd*, *Vhu*, *Vhc*, *Ehb* and *Eha*, as an inorganic electron carrier (Lie et al., 2012; Costa et al., 2013). Table 5.2 shows that *Frc* and *Fru* and *Vhu* and *Vhc* were found to be mapped to the same K numbers, respectively. To avoid redundancy, these were treated as one enzyme, represented by *Frc/Fru* and *Vhc/Vhu*. As Croese et al. (2011) proposed that Dv could perform indirect electron transfer by releasing organic molecules such as c-type cytochromes and *Fdh*, Mm's genome was searched for genes annotated with these compounds. Five genes were found to be annotated with *Fdh* subunit genes. However, there was some redundancy and these were mapped to 3 distinct subunit and K numbers, which were used in the analysis. A c-type cytochrome biogenesis protein (*Cyc*) was also found to be annotated in Mm's genome and mapped to a K number, which was also included in the analysis.

**Table 5.1** Mapping of Dv's genes potentially involved in electron transfer to K numbers.

	Protein	Gene	Gene ID	K number	K number name	EC number
1	Coo	<i>cooM</i>	DVU2286			
2		<i>cooK</i>	DVU2287			
3		<i>cooL</i>	DVU2288			
4		<i>cooX</i>	DVU2289			
5		<i>cooU</i>	DVU2290			
6		<i>cooH</i>	DVU2291			
7		<i>cooF</i>	DVU2293	K00196	anaerobic carbon-monoxide dehydrogenase iron sulphur subunit	
8		<i>cooS</i>	DVU2098	K00198	anaerobic carbon-monoxide dehydrogenase catalytic subunit	1.2.7.4
9	Hmc	<i>hmc</i>	DVU0531			
10		<i>hmc</i>	DVU0532			
11		<i>hmc</i>	DVU0533			
12		<i>hmc</i>	DVU0534			
13		<i>hmc</i>	DVU0535			
14		<i>hmc</i>	DVU0536			
15	Hyn	<i>hynB-1</i>	DVU1921	K18008	[NiFe] hydrogenase small subunit	1.12.2.1
16		<i>hynA-1</i>	DVU1922	K00437	[NiFe] hydrogenase large subunit	1.12.2.1
17	Hyd	<i>hydA</i>	DVU1769	K00533	ferredoxin hydrogenase large subunit	1.12.7.2
18		<i>hydB</i>	DVU1770	K00534	ferredoxin hydrogenase small subunit	1.12.7.2
19	Ccm	<i>ccmC</i>	DVU1047	K02195	heme exporter protein	
20		<i>ccmB</i>	DVU1048	K02194	heme exporter protein	
21		<i>ccmF</i>	DVU1050	K02198	cytochrome c-type biogenesis protein	
22		<i>ccmE</i>	DVU1051	K02197	cytochrome c-type biogenesis protein	
23	Flg	<i>flgE</i>	DVU0307	K02390	flagellar hook protein FlgE	
24		<i>flgC</i>	DVU0315	K02388	flagellar basal-body rod protein FlgC	

*Continued on next page*



Table 5.1 – continued from previous page

	Protein	Gene	Gene ID	K number	K number name	EC number
25		<i>flgB</i>	DVU0316	K02387	flagellar basal-body rod protein FlgB	
26		<i>flgG</i>	DVU0512	K02392	flagellar basal-body rod protein FlgG	
27		<i>flgG</i>	DVU0513	K02392	flagellar basal-body rod protein FlgG	
28		<i>flgA</i>	DVU0514	K02386	flagella basal body P-ring formation protein FlgA	
29		<i>flgH</i>	DVU0515	K02393	flagellar L-ring protein precursor FlgH	
30		<i>flgI</i>	DVU0516	K02394	flagellar P-ring protein precursor FlgI	
31		<i>flgK</i>	DVU0519	K02396	flagellar hook-associated protein 1 FlgK	
32		<i>flgL</i>	DVU0520	K02397	flagellar hook-associated protein 3 FlgL	
33		<i>flgM</i>	DVU0523	K02398	negative regulator of flagellin synthesis FlgM	
34		<i>flgE</i>	DVU1443	K02390	flagellar hook protein FlgE	
35		<i>flgD</i>	DVU1444	K02389	flagellar basal-body rod modification protein FlgD	
36		<i>flgC</i>	DVU2893	K02388	flagellar basal-body rod protein FlgC	
37	Fdh	<i>fdhE</i>	DVU0577	K02380	FdhE protein	
38		<i>fdhD</i>	DVU0578	K02379	FdhD protein	
39		<i>fdhE</i>	DVU2810	K02380	FdhE protein	

Table 5.2 Mapping of Mm's genes potentially involved in electron transfer to K numbers.

	Protein	Gene	Gene ID	K number	K number name	EC number
1	Fru	<i>fruA</i>	MMP1382	K00440	coenzyme F420 hydrogenase subunit alpha	1.12.98.1
2		<i>fruD</i>	MMP1383	K00442	coenzyme F420 hydrogenase subunit delta	1.12.98.1
3		<i>fruG</i>	MMP1384	K00443	coenzyme F420 hydrogenase subunit gamma	1.12.98.1
4		<i>fruB</i>	MMP1385	K00441	coenzyme F420 hydrogenase subunit beta	1.12.98.1
5	Frc	<i>frcB</i>	MMP0817	K00441	coenzyme F420 hydrogenase subunit beta	1.12.98.1
6		<i>frcG</i>	MMP0818	K00443	coenzyme F420 hydrogenase subunit gamma	1.12.98.1

Continued on next page

Table 5.2 – continued from previous page

	<b>Protein</b>	<b>Gene</b>	<b>Gene ID</b>	<b>K number</b>	<b>K number name</b>	<b>EC number</b>
7		<i>frcD</i>	MMP0819	K00442	coenzyme F420 hydrogenase subunit delta	1.12.98.1
8		<i>frcA</i>	MMP0820	K00440	coenzyme F420 hydrogenase subunit alpha	1.12.98.1
9	Hmd	<i>hmd</i>	MMP0127	K13942	5,10-methenyltetrahydromethanopterin hydrogenase	1.12.98.2
10	Vhu	<i>vhuA</i>	MMP1694	K14126	F420-non-reducing hydrogenase large subunit	1.8.98.5
11		<i>vhuB</i>	MMP1692			
12		<i>vhuG</i>	MMP1695	K14128	F420-non-reducing hydrogenase small subunit	1.8.98.5
13		<i>vhuD</i>	MMP1696	K14127	F420-non-reducing hydrogenase iron-sulphur subunit	1.8.98.5
14		<i>vhuU</i>	MMP1693			
15	Vhc	<i>vhcA</i>	MMP0823	K14126	F420-non-reducing hydrogenase large subunit	1.8.98.5
16		<i>vhcG</i>	MMP0822	K14128	F420-non-reducing hydrogenase small subunit	1.8.98.5
17		<i>vhcD</i>	MMP0821	K14127	F420-non-reducing hydrogenase iron-sulphur subunit	1.8.98.5
18	Ehb	<i>ehbQ</i>	MMP0400	K06862	energy-converting hydrogenase B subunit Q	
19		<i>ehbP</i>	MMP0940	K14125	energy-converting hydrogenase B subunit P	
20		<i>ehbB</i>	MMP1049	K14111	energy-converting hydrogenase B subunit B	
21		<i>ehbC</i>	MMP1073	K14112	energy-converting hydrogenase B subunit C	
22		<i>ehbD</i>	MMP1074	K14113	energy-converting hydrogenase B subunit D	
23		<i>ehbN</i>	MMP1153	K14123	energy-converting hydrogenase B subunit N	
24		<i>ehbA</i>	MMP1469	K14110	energy-converting hydrogenase B subunit A	
25		<i>ehbO</i>	MMP1621	K14124	energy-converting hydrogenase B subunit O	
26		<i>ehbM</i>	MMP1622	K14122	energy-converting hydrogenase B subunit M	
27		<i>ehbL</i>	MMP1623	K14121	energy-converting hydrogenase B subunit L	
28		<i>ehbK</i>	MMP1624	K14120	energy-converting hydrogenase B subunit K	
29		<i>ehbJ</i>	MMP1625	K14119	energy-converting hydrogenase B subunit J	
30		<i>ehbG</i>	MMP1627	K14116	energy-converting hydrogenase B subunit G	
31		<i>ehbF</i>	MMP1628	K14115	energy-converting hydrogenase B subunit F	

Continued on next page

Table 5.2 – continued from previous page

	<b>Protein</b>	<b>Gene</b>	<b>Gene ID</b>	<b>K number</b>	<b>K number name</b>	<b>EC number</b>
32		<i>ehbE</i>	MMP1629	K14114	energy-converting hydrogenase B subunit E	
33	Eha	<i>ehaA</i>	MMP1448	K14092	energy-converting hydrogenase A subunit A	
34		<i>ehaB</i>	MMP1449	K14093	energy-converting hydrogenase A subunit B	
35		<i>ehaC</i>	MMP1450	K14094	energy-converting hydrogenase A subunit C	
36		<i>ehaD</i>	MMP1451	K14095	energy-converting hydrogenase A subunit D	
37		<i>ehaE</i>	MMP1452	K14096	energy-converting hydrogenase A subunit E	
38		<i>ehaF</i>	MMP1453	K14097	energy-converting hydrogenase A subunit F	
39		<i>ehaG</i>	MMP1454	K14098	energy-converting hydrogenase A subunit G	
40		<i>ehaH</i>	MMP1455	K14099	energy-converting hydrogenase A subunit H	
41		<i>ehaI</i>	MMP1456	K14100	energy-converting hydrogenase A subunit I	
42		<i>ehaJ</i>	MMP1457	K14101	energy-converting hydrogenase A subunit J	
43		<i>ehaK</i>	MMP1458	K14102	energy-converting hydrogenase A subunit K	
44		<i>ehaL</i>	MMP1459	K14103	energy-converting hydrogenase A subunit L	
45		<i>ehaM</i>	MMP1460	K14104	energy-converting hydrogenase A subunit M	
46		<i>ehaN</i>	MMP1461	K14105	energy-converting hydrogenase A subunit N	
47		<i>ehaO</i>	MMP1462	K14106	energy-converting hydrogenase A subunit O	
48		<i>ehaP</i>	MMP1463	K14107	energy-converting hydrogenase A subunit P	
49	Fdh	<i>fdhA</i>	MMP0138	K22516	formate dehydrogenase alpha subunit	1.17.98.3, 1.8.98.6
50		<i>fdhB</i>	MMP0139	K00125	formate dehydrogenase beta subunit	1.17.98.3, 1.8.98.6
51		<i>fdhD</i>	MMP1233	K02379	FdhD protein	
52		<i>fdhA</i>	MMP1298	K22516	formate dehydrogenase alpha subunit	1.17.98.3, 1.8.98.6
53		<i>fdhB</i>	MMP1297	K00125	formate dehydrogenase beta subunit	1.17.98.3, 1.8.98.6
54	Cyc	<i>cycZ</i>	MMP0957	K06196	cytochrome c-type biogenesis protein	

**Table 5.3** Summary of KEGG genomes identified using the anodic protein search.

Protein	K numbers	No. genomes matched	No. unique organisms	% KEGG genome database
Ccm	K02198,K02197	1698	1688	32.19
Coo	K00196,K00198	340	334	6.48
Fdh	K02380,K02379	2646	2632	50.46
Flg	K02390,K02388,K02387,K02392, K02386,K02393,K02394,K02396 K02397,K02398,K02389	2339	2322	44.60
Hyd	K00533,K00534	48	48	0.92
Hyn	K18008,K00437	28	28	0.53

### 5.2.1 Use of the mapped K numbers to identify KEGG genomes

Six proteins were used as a search query in order to identify anodic organisms (Ccm, Coo, Fdh, Flg, Hyd and Hyn), while seven sets of proteins were used as a search query in order to identify cathodic organisms (Cyc, Eha, Ehb, Fdh, Frc/Fru, Hmd, and Vhc/Vhu). These are referred as anodic and cathodic protein searches, respectively, throughout this chapter.

The K numbers for each protein set were used to find the KEGG genomes annotated with the those K numbers using the MetQy's `query/genes/to/genomes` function (see Code 5.1). In order to accommodate limited or poor annotations, a single protein subunit was required to be annotated in the KEGG genomes for that protein to be considered to be encoded in the genome. Table 5.3 summarises the number of KEGG genomes and unique organisms matched for the anodic protein search. Given that MetQy's KEGG genome database has 5,244 entries, *fdh* and *flg* have been annotated in 50.46 and 44.6 % of the genomes, reflecting how widespread the gene annotation is. *ccm* is also relatively frequent, as it has been annotated in 32.53 % of KEGG genomes. On the other hand, only 0.53, 0.92 and 6.48 % of KEGG genomes have been annotated with *hyn*, *hyd* and *coo*, respectively. During the use of MetQy's `query/genes/to/genomes` function, a warning was issued stating that the K number K22516 (*fdhA*, formate dehydrogenase alpha subunit) is not a valid KEGG ortholog ID. This was due to the fact that this particular K number was added to

**Table 5.4** Summary of KEGG genomes identified using the cathodic protein search.

Protein	K numbers	No. genomes matched	No. unique organisms	% KEGG genome database
Cyc	K06196	1738	1722	33.14
Eha	K14092,K14093,K14094,K14095, K14096,K14097,K14098,K14099, K14100,K14101,K14102,K14103, K14104,K14105,K14106,K14107	41	40	0.78
Ehb	K06862,K14125,K14111,K14112, K14113,K14123,K14110,K14124, K14122,K14121,K14120,K14119, K14116,K14115,K14114	63	63	1.20
Fdh	K22516,K00125,K02379	2598	2584	49.54
Frc/Fru	K00440,K00442,K00443,K00441	217	217	4.14
Hmd	K13942	25	25	0.48
Vhc/Vhu	K14126,K14128,K14127	124	124	2.36

**Table 5.5** Co-occurrence frequency of genomes annotated with the anodic proteins.

	Co-occurrence	Freq		Co-occurrence	Freq
1	Ccm, Fdh, Flg	965	16	Coo, Fdh, Flg, Hyn	6
2	Fdh, Flg	610	17	Ccm, Coo, Fdh, Flg, Hyn	5
3	Fdh	571	18	Ccm, Coo, Flg	5
4	Flg	336	19	Coo, Fdh, Flg, Hyd	5
5	Ccm, Fdh	277	20	Fdh, Flg, Hyd	3
6	Ccm, Flg	200	21	Flg, Hyd	3
7	Ccm	181	22	Coo, Fdh, Hyd, Hyn	2
8	Coo, Flg	94	23	Coo, Flg, Hyd	2
9	Coo, Fdh	75	24	Ccm, Coo	1
10	Coo, Fdh, Flg	57	25	Ccm, Coo, Fdh, Hyd, Hyn	1
11	Coo	33	26	Ccm, Coo, Flg, Hyd	1
12	Ccm, Coo, Fdh	23	27	Ccm, Fdh, Hyd	1
13	Ccm, Coo, Fdh, Flg	16	28	Ccm, Flg, Hyd	1
14	Ccm, Fdh, Flg, Hyd	16	29	Coo, Flg, Hyn	1
15	Ccm, Coo, Fdh, Flg, Hyd, Hyn	13			

the KEGG ortholog database after the last update of MetQy's in-built data (July 2018).

Table 5.4 summarises the number of KEGG genomes and unique organisms matched for each of Mm's proteins. *fdh* and *cyc* were annotated in 49.54 and 33.14 % of the KEGG genome entries, the most frequent for this set of proteins. All other genes were annotated in less than 5 % of the KEGG genome entries.

### 5.2.2 Identification of protein co-annotation across genomes

Of course, the genomes can be annotated with more than one protein. Tables 5.5 and 5.6 summarise the protein co-occurrence across KEGG genomes for anodic and cathodic protein searches, respectively. A total of 3,504 distinct KEGG genomes were retrieved based on Dv's proteins and these occurred in 29 protein combinations. Mm's protein led to 3,345 distinct KEGG genomes being identified in 32 protein combinations. The most frequent protein annotation based on Mm's protein search was Fdh alone (1,470 genomes). Interestingly, the most common protein

**Table 5.6** Co-occurrence frequency of genomes annotated with the cathodic proteins.

	Co-occurrence	Freq		Co-occurrence	Freq
1	Fdh	1470	17	Fdh, Vhc/Vhu	5
2	Cyc, Fdh	942	18	Eha, Ehb, Fdh, Frc/Fru, Vhc/Vhu	4
3	Cyc	664	19	Ehb, Fdh, Frc/Fru	4
4	Cyc, Fdh, Frc/Fru	44	20	Ehb, Fdh, Vhc/Vhu	4
5	Fdh, Frc/Fru	40	21	Cyc, Ehb, Fdh, Frc/Fru	3
6	Frc/Fru	30	22	Cyc, Vhc/Vhu	3
7	Vhc/Vhu	17	23	Cyc, Eha	2
8	Cyc, Fdh, Vhc/Vhu	16	24	Cyc, Ehb, Fdh, Vhc/Vhu	2
9	Cyc, Frc/Fru	16	25	Cyc, Eha, Ehb, Fdh, Frc/Fru	1
10	Cyc, Frc/Fru, Vhc/Vhu	13	26	Cyc, Ehb, Fdh	1
11	Eha, Ehb, Fdh, Frc/Fru, Hmd, Vhc/Vhu	13	27	Eha, Ehb, Fdh, Frc/Fru, Hmd	1
12	Cyc, Eha, Ehb, Fdh, Frc/Fru, Vhc/Vhu	11	28	Ehb	1
13	Cyc, Fdh, Frc/Fru, Vhc/Vhu	11	29	Ehb, Fdh	1
14	Cyc, Eha, Ehb, Fdh, Frc/Fru, Hmd, Vhc/Vhu	9	30	Ehb, Fdh, Frc/Fru, Hmd	1
15	Fdh, Frc/Fru, Vhc/Vhu	8	31	Ehb, Fdh, Frc/Fru, Hmd, Vhc/Vhu	1
16	Ehb, Fdh, Frc/Fru, Vhc/Vhu	6	32	Frc/Fru, Vhc/Vhu	1

co-occurrence based on Dv's proteins is Ccm, Fdh and Flg (965 genomes) and Fdh alone was the third most common with 571 genomes. It is also evident in both tables (Tables 5.5 and 5.6) that Fdh is widely spread, as they occur in most protein combinations. Flg also appears frequently in the protein combinations listed in Table 5.5.

A closer look was obtained by aggregating the protein co-occurrence annotated by phyla (Tables 5.7 and 5.8). In both cases, the top four were Proteobacteria, Firmicutes, Actinobacteria and Euryarchaeota with (2022, 1677) (580, 614) (398, 496) (114, 144) for (Dv, Mm), respectively. The fifth phyla did differ and it was Bacteroidetes with 81 genomes and Cyanobacteria with 103 genomes for anodic and cathodic protein searches, respectively.

**Table 5.7** Protein co-occurrence for each phylum based on Dv's protein search.

	Phyla	Total	Co-occurrence
1	Acidobacteria	8	Ccm,Fdh (1), Ccm,Fdh,Flg (4), Ccm,Flg (3)
2	Actinobacteria	398	Ccm,Fdh (16), Ccm,Fdh,Flg (5), Fdh (337), Fdh,Flg (27), Flg (13)
3	Aquificae	14	Coo,Fdh,Flg,Hyd (1), Coo,Flg (3), Fdh (2), Fdh,Flg (3), Flg (5)
4	Armatimonadetes	2	Ccm,Flg (2)
5	Asgard group	1	Ccm,Fdh (1)
6	Bacteroidetes	81	Ccm (27), Ccm,Fdh (28), Ccm,Fdh,Flg (1), Ccm,Flg (4), Coo (1), Fdh (20)
7	Caldiserica	1	Fdh (1)
8	Calditrichaeota	1	Ccm,Flg (1)
9	candidate division NC10	1	Ccm,Fdh,Flg (1)
10	Candidatus Bathyarchaeota	1	Coo,Fdh (1)
11	Candidatus Korarchaeota	1	Coo,Fdh (1)
12	Candidatus Melainabacteria	1	Flg (1)
13	Chlorobi	4	Ccm (1), Coo (3)
14	Chloroflexi	26	Ccm (2), Ccm,Coo,Fdh (1), Ccm,Fdh (7), Ccm,Fdh,Flg (2), Coo (1), Coo,Fdh (13)
15	Chrysiogenetes	1	Coo,Flg (1)
16	Crenarchaeota	46	Ccm (5), Ccm,Fdh (7), Coo,Fdh (1), Fdh (33)
17	Cyanobacteria	16	Fdh (9), Fdh,Flg (3), Flg (4)
18	Deferribacteres	4	Ccm,Fdh (1), Ccm,Fdh,Flg (1), Ccm,Flg (2)
19	Deinococcus-Thermus	27	Ccm (8), Ccm,Fdh (19)
20	Dictyoglomi	1	Coo (1)
21	Elusimicrobia	2	Coo (2)
22	Euryarchaeota	114	Ccm (7), Ccm,Coo,Fdh (22), Ccm,Fdh (8), Coo (12), Coo,Fdh (52), Fdh (13)
23	Firmicutes	580	Ccm,Coo,Fdh,Flg (8), Ccm,Coo,Flg (2), Ccm,Coo,Flg,Hyd (1), Ccm,Fdh,Flg (17), Ccm,Flg (1), Ccm,Flg,Hyd (1), Coo (11), Coo,Fdh (2), Coo,Fdh,Flg (44), Coo,Fdh,Flg,Hyd (3), Coo,Flg (70), Coo,Flg,Hyd (2), Fdh (95), Fdh,Flg (248), Fdh,Flg,Hyd (1), Flg (71), Flg,Hyd (3)

*Continued on next page*

Table 5.7 – continued from previous page

	Phyla	Total	Co-occurrence
24	Fusobacteria	2	Fdh (2)
25	Gemmatimonadetes	3	Ccm,Fdh,Flg (2), Ccm,Flg (1)
26	Ignavibacteriae	2	Ccm,Flg (2)
27	Nitrospirae	8	Ccm,Flg (1), Fdh,Flg (4), Fdh,Flg,Hyd (1), Flg (2)
28	Planctomycetes	16	Fdh,Flg (5), Flg (11)
29	Proteobacteria	2022	Ccm (131), Ccm,Coo (1), Ccm,Coo,Fdh,Flg (8), Ccm,Coo,Fdh,Flg,Hyd,Hyn (13), Ccm,Coo,Fdh,Flg,Hyn (5), Ccm,Coo,Fdh,Hyn (1), Ccm,Coo,Flg (3), Ccm,Fdh (188), Ccm,Fdh,Flg (932), Ccm,Fdh,Hyd (16), Ccm,Fdh,Hyd (1), Ccm,Flg (172), Coo,Fdh (4), Coo,Fdh,Flg (12), Coo,Fdh,Flg,Hyn (5), Coo,Fdh,Hyd,Hyn (2), Coo,Flg (4), Coo,Flg,Hyn (1), Fdh (55), Fdh,Flg (316), Fdh,Flg,Hyd (1), Flg (151)
30	Spirochaetes	76	Ccm,Flg (11), Coo (2), Coo,Fdh,Flg (1), Coo,Flg (2), Flg (60)
31	Synergistetes	5	Coo,Fdh (1), Fdh,Flg (1), Flg (3)
32	Thaumarchaeota	1	Fdh (1)
33	Thermobaculum	1	Ccm,Fdh (1)
34	Thermodesulfobacteria	4	Coo,Fdh,Flg,Hyd (1), Coo,Fdh,Flg,Hyn (1), Coo,Flg (1), Fdh,Flg (1)
35	Thermotogae	27	Coo,Flg (13), Fdh,Flg (1), Flg (13)
36	Verrucomicrobia	6	Fdh (3), Fdh,Flg (1), Flg (2)

Table 5.8 Protein co-occurrence for each phylum based on Mm’s protein search.

	Phyla	Total	Co-occurrence
1	Acidobacteria	7	Cyc (2), Ehb (1), Ehb,Fdh (1), Fdh (2), Frc/Fru (1)
2	Actinobacteria	496	Cyc (112), Cyc,Fdh (366), Cyc,Fdh,Frc/Fru (14), Ehb,Fdh,Frc/Fru (2), Fdh (2)
3	Aquificae	14	Cyc (9), Cyc,Fdh (2), Cyc,Fdh,Vhc/Vhu (1), Fdh,Frc/Fru,Vhc/Vhu (1), Frc/Fru (1)
4	Asgard group	1	Cyc,Fdh,Frc/Fru,Vhc/Vhu (1)
5	Bacteroidetes	52	Cyc (1), Cyc,Fdh (1), Fdh (48), Vhc/Vhu (2)

Continued on next page



Table 5.8 – continued from previous page

	Phyla	Total	Co-occurrence
6	Caldiserica	1	Cyc,Fdh (1)
7	Calditrichaeota	1	Vhc/Vhu (1)
8	candidate division NC10	1	Cyc,Fdh (1)
9	Candidatus Bathyarchaeota	2	Fdh,Frc/Fru,Vhc/Vhu (1), Vhc/Vhu (1)
10	Candidatus Korarchaeota	1	Vhc/Vhu (1)
11	Candidatus Nanohaloarchaeota	1	Cyc (1)
12	Candidatus Woesebacteria	1	Cyc (1)
13	Candidatus Wolfebacteria	1	Cyc (1)
14	Chloroflexi	25	Cyc (3), Cyc,Fdh (7), Cyc,Fdh,Frc/Fru,Vhc/Vhu (2), Cyc,Frc/Fru,Vhc/Vhu (12), Frc/Fru,Vhc/Vhu (1)
15	Chrysiogenetes	1	Cyc (1)
16	Crenarchaeota	43	Cyc,Fdh (3), Fdh (36), Fdh,Vhc/Vhu (1), Frc/Fru (1), Vhc/Vhu (2)
17	Cyanobacteria	103	Cyc (91), Cyc,Fdh (11), Cyc,Fdh,Frc/Fru (1)
18	Deferribacteres	4	Cyc (2), Cyc,Fdh (2)
19	Deinococcus-Thermus	27	Cyc (8), Cyc,Fdh (19)
20	Euryarchaeota	144	Cyc (17), Cyc,Eha,Ehb,Fdh,Frc/Fru (1), Cyc,Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu (9), Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu (11), Cyc,Ehb,Fdh,Frc/Fru (3), Cyc,Ehb,Fdh,Vhc/Vhu (2), Cyc,Fdh,Frc/Fru (8), Cyc,Frc/Fru (14), Cyc,Vhc/Vhu (1), Eha,Ehb,Fdh,Frc/Fru,Hmd (1), Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu (13), Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu (4), Ehb,Fdh,Frc/Fru (2), Ehb,Fdh,Frc/Fru,Hmd (1), Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu (1), Ehb,Fdh,Frc/Fru,Vhc/Vhu (6), Ehb,Fdh,Vhc/Vhu (4), Fdh (6), Fdh,Frc/Fru (23), Fdh,Frc/Fru,Vhc/Vhu (3), Fdh,Vhc/Vhu (1), Frc/Fru (8), Vhc/Vhu (5), Cyc (192), Cyc,Eha (2), Cyc,Fdh (256), Cyc,Fdh,Frc/Fru (2), Cyc,Fdh,Frc/Fru,Vhc/Vhu (2), Cyc,Fdh,Vhc/Vhu (1), Cyc,Frc/Vhu (1), Cyc,Frc/Fru (2), Fdh,Vhc/Vhu (3), Frc/Fru (2), Vhc/Vhu (1), Cyc (11), Fdh (2)
21	Firmicutes	614	
22	Fusobacteria	13	Cyc (11), Fdh (2)
23	Gemmatimonadetes	3	Cyc (1), Cyc,Ehb,Fdh (1), Cyc,Fdh (1)
24	Ignavibacteriae	1	Cyc (1)
25	Nitrospirae	9	Cyc (8), Cyc,Fdh,Vhc/Vhu (1)

Continued on next page

Table 5.8 – continued from previous page

	Phyla	Total	Co-occurrence
26	Planctomycetes	8	Cyc (2), Fdh (5), Vhc/Vhu (1)
27	Proteobacteria	1677	Cyc (121), Cyc,Fdh (268), Cyc,Fdh,Frc/Fru (19), Cyc,Fdh,Frc/Fru,Vhc/Vhu (6), Cyc,Fdh,Vhc/Vhu (12), Cyc,Frc/Fru,Vhc/Vhu (1), Cyc,Vhc/Vhu (2), Fdh (1216), Fdh,Frc/Fru (13), Fdh,Frc/Fru,Vhc/Vhu (2), Frc/Fru (16), Vhc/Vhu (1)
28	Spirochaetes	23	Cyc (22), Cyc,Fdh (1)
29	Stramenopiles	1	Frc/Fru (1)
30	Synergistetes	5	Cyc (3), Cyc,Fdh (2)
31	Thaumarchaeota	15	Cyc (14), Fdh,Frc/Fru (1)
32	Thermobaculum	1	Fdh,Frc/Fru (1)
33	Thermodesulfobacteria	4	Fdh (1), Fdh,Frc/Fru,Vhc/Vhu (1), Vhc/Vhu (2)
34	Thermotogae	29	Cyc (28), Cyc,Fdh (1)
35	UNKNOWN	1	Cyc (1)
36	Verrucomicrobia	4	Cyc,Fdh,Vhc/Vhu (1), Fdh (3)
37	Viridiplantae	11	Cyc (11)

### 5.2.3 Method evaluation: determination of expected identified genomes

To evaluate the effectiveness of this search, the identification of certain organisms was verified. Both Dv and Mm have been annotated with K02379 (*fdhD*), which would suggest that Mm would be identified in the anodic protein search and Dv would be identified in the cathodic protein search. Therefore, the identification of *Methanococcus* and *Desulfovibrio* genomes was checked.

Mm (*Methanococcus maripaludis* sp.S2) was identified during the anodic protein search, as well as the species *Methanococcus aeolicus* sp. Nankai-3, *Methanococcus maripaludis* sp. C5, C6, C7, and X1, *Methanococcus vanniellii* sp. SB and *Methanococcus voltae* sp. A3. All 8 genomes were annotated with both Coo and Fdh. Eleven *Desulfovibrio* genomes were identified, including Dv's (*Desulfovibrio vulgaris* Hildenborough), which are shown in Table 5.9.

**Table 5.9** Protein co-occurrence of the genus *Desulfovibrio* found when searching for Mm's proteins

Genome ID	Organism	Co-occurrence
T01662	<i>Desulfovibrio africanus</i> Walvis Bay	Cyc,Fdh,Frc/Fru,Vhc/Vhu
T00294	<i>Desulfovibrio alaskensis</i> G20 ( <i>Desulfovibrio desulfuricans</i> G20)	Cyc,Fdh
T01709	<i>Desulfovibrio desulfuricans</i> ND132	Cyc,Fdh
T00837	<i>Desulfovibrio desulfuricans</i> subsp. <i>desulfuricans</i> ATCC 27774	Fdh
T04293	<i>Desulfovibrio fairfieldensis</i> CCUG 45958	Fdh
T02851	<i>Desulfovibrio gigas</i> DSM 1382 = ATCC 19364	Fdh
T02428	<i>Desulfovibrio hydrothermalis</i> AM13 = DSM 14728	Fdh
T00917	<i>Desulfovibrio magneticus</i> RS-1	Cyc,Fdh
T00969	<i>Desulfovibrio salexigens</i> DSM 2638	Cyc,Fdh
T05183	<i>Desulfovibrio</i> sp. G11	Fdh
T00452	<i>Desulfovibrio vulgaris</i> DP4	Fdh
T00171	<i>Desulfovibrio vulgaris</i> Hildenborough	Fdh
T00816	<i>Desulfovibrio vulgaris</i> Miyazaki F	Fdh
T02064	<i>Desulfovibrio vulgaris</i> RCH1	Fdh
T01382	<i>Pseudodesulfovibrio aespoeensis</i> ( <i>Desulfovibrio aespoeensis</i> ) Aspo-2	Cyc,Fdh,Vhc/Vhu
T04404	<i>Pseudodesulfovibrio indicus</i> ( <i>Desulfovibrio indicus</i> ) J2	Cyc,Fdh,Vhc/Vhu
T02481	<i>Pseudodesulfovibrio piezophilus</i> ( <i>Desulfovibrio piezophilus</i> ) C1TLV30	Cyc,Fdh

The cross-organism identification validated the protein searches using the potential mechanisms. Additionally, the retrieval on known electroactive microorganisms of the genera *Geobacter* and *Shewanella* was evaluated to further demonstrate that the anodic and cathodic protein searches could lead to the identification of unknown electroactive organisms. Table 5.10 shows that 11 *Geobacter* and 28 *Shewanella* genomes were identified with this analysis. In the anodic search

**Table 5.10** Protein co-occurrence of known electroactive genera.

Anodic Co-occurrence	<i>Geobacter</i>	<i>Shewanella</i>
Ccm, Fdh, Flg	0	22
Ccm, Fdh, Flg, Hyd	0	4
Ccm, Flg	0	2
Coo, Fdh, Flg	6	0
Fdh, Flg	2	0
Coo, Fdh, Flg, Hyn	1	0
Coo, Flg	1	0
Flg	1	0
<b>Anodic TOTAL</b>	<b>11</b>	<b>28</b>

Cathodic Co-occurrence	<i>Geobacter</i>	<i>Shewanella</i>
Fdh	0	26
Cyc,Fdh,Vhc/Vhu	5	0
Cyc,Fdh	4	0
Cyc,Vhc/Vhu	2	0
<b>Cathodic TOTAL</b>	<b>11</b>	<b>26</b>

**Table 5.11** Electroactive organisms identified by both the anodic and cathodic protein searches.

	Genome ID	Organism	Reported as electroactive (e.g.)
1	T04399	<i>Geobacter anodireducens</i> SD-1	Sun et al. (2014)
2	T00749	<i>Geobacter bemidjiensis</i> Bem	Butler et al. (2010)
3	T00846	<i>Geobacter daltonii</i> FRC-32	Butler et al. (2010)
4	T00708	<i>Geobacter lovleyi</i> SZ	Butler et al. (2010)
5	T00295	<i>Geobacter metallireducens</i> GS-15	Butler et al. (2010)
6	T03568	<i>Geobacter pickeringii</i> G13	
7	T01421	<i>Geobacter</i> sp. M18	
8	T00933	<i>Geobacter</i> sp. M21	
9	T01940	<i>Geobacter sulfurreducens</i> KN400	Kracke (2016)
10	T00155	<i>Geobacter sulfurreducens</i> PCA	Butler et al. (2010); Liu et al. (2014)
11	T00521	<i>Geobacter uraniireducens</i> Rf4	Butler et al. (2010)
12	T00445	<i>Shewanella amazonensis</i> SB2B	
13	T02014	<i>Shewanella baltica</i> BA175	
14	T02013	<i>Shewanella baltica</i> OS117	
15	T00478	<i>Shewanella baltica</i> OS155	
16	T00571	<i>Shewanella baltica</i> OS185	
17	T00622	<i>Shewanella baltica</i> OS195	
18	T00803	<i>Shewanella baltica</i> OS223	
19	T01725	<i>Shewanella baltica</i> OS678	
20	T05197	<i>Shewanella bicestria</i> JAB-1	
21	T00395	<i>Shewanella frigidimarina</i> NCIMB 400	
22	T00654	<i>Shewanella halifaxensis</i> HAW-EB4	
23	T04959	<i>Shewanella japonica</i> KCTC 22435	Biffinger et al. (2011) <sup>1</sup>
24	T00489	<i>Shewanella loihica</i> PV-4	Jain et al. (2012); Zhang et al. (2015)
25	T00099	<i>Shewanella oneidensis</i> MR-1	Pirbadian et al. (2014); Miller et al. (2016); Hong and Pachter (2016); Zhu et al. (2017)*
26	T00606	<i>Shewanella pealeana</i> ATCC 700345	
27	T00791	<i>Shewanella piezotolerans</i> WP3	
28	T05063	<i>Shewanella psychrophila</i> WP2	
29	T02015	<i>Shewanella putrefaciens</i> 200	Kim et al. (2002); Wu et al. (2014)
30	T00513	<i>Shewanella putrefaciens</i> CN-32	
31	T00596	<i>Shewanella sediminis</i> HAW-EB3	
32	T00428	<i>Shewanella</i> sp. ANA-3	
33	T04944	<i>Shewanella</i> sp. FDAARGOS/354	
34	T00388	<i>Shewanella</i> sp. MR-4	
35	T00389	<i>Shewanella</i> sp. MR-7	
36	T00453	<i>Shewanella</i> sp. W3-18-1	
37	T00676	<i>Shewanella woodyi</i> ATCC 51908	Tian et al. (2017)

<sup>1</sup> Identified at species level

\* to name a few

outcome, all *Shewanella* genomes were annotated with *ccm* and *flg*, while all *Geobacter* genomes were annotated with *flg* and most with *coo*. In the cathodic search, all *Shewanella* genomes were annotated with *fdh* and all *Geobacter* genomes with *cyc* and either *fdh* or *vhc/vhu* or both. Surprisingly, however, the *Shewanella* and *Geobacter* genomes identified with both searches were the same (see Table 5.11), except that the anodic search identified two additional *Shewanella* genomes: *Shewanella denitrificans* OS217 and *Shewanella violacea* DSS12. Table 5.11 also includes studies where the organisms have demonstrated EET.

### 5.2.4 Selecting genome subsets through different filtering methods

Given that 3,504 and 3,345 distinct KEGG genomes were retrieved for the anodic and cathodic protein searches, respectively, a subset of these needed to be identified to guide the experimental

design of future “syntrophy over wires” hypothesis tests. The information gathered could be used to narrow the selection systematically using various criteria, including the use of taxonomic information, the protein [co-]occurrence and qualitative biochemical information obtained with MetQy functions. Ultimately, the final choice would be constrained by the available organisms and literature searches to corroborate any information found through KEGG.

The protein co-occurrence could be used to restrict the proteins required to be annotated. For example, the decision could be made that future experiments to test the “syntrophy over wires” hypothesis should be carried out using an anodic organism that has the same six proteins as Dv and a cathodic organism that has the same seven proteins as Mm. Therefore, according to Table 5.5, there are 13 and 9 genomes annotated with all of Dv’s and Mm’s proteins, respectively, which are listed in Table 5.12. Not surprisingly, Dv (*Desulfovibrio vulgaris* Hildenborough) is listed as an anodic organism, together with three other subspecies and nine other members of the same genus. Consistently, Mm (*Methanococcus maripaludis* S2) is also listed as a possible cathodic organism, as well as 2 other different subspecies and another *Methanococcus* species. Although all cathodic organisms are from the Euryarchaeota phylum, there is a larger genus-level diversity than for anodic organisms. These genera include: *Methanobrevibacter*, *Methanocaldococcus*, *Methanothermobacter*, *Methanotorris* and *Methanococcus*. This would result in  $(13 \times 9) - 1 = 116$  combinations of anodic and cathodic organisms. One has been subtracted as it would represent the pairing of Dv with Mm.

Another approach to selecting organisms would be to use taxonomic information at different

**Table 5.12** Workable experiment organisms based on the anodic and cathodic protein searches.

		Genome ID	Phylum	Organism
1	Anodic	T00294	Proteobacteria	<i>Desulfovibrio alaskensis</i> ( <i>Desulfovibrio desulfuricans</i> G20)
2		T00837	Proteobacteria	<i>Desulfovibrio desulfuricans</i> subsp. <i>desulfuricans</i> ATCC 27774
3		T04293	Proteobacteria	<i>Desulfovibrio fairfieldensis</i> CCUG 45958
4		T02428	Proteobacteria	<i>Desulfovibrio hydrothermalis</i> AM13 = DSM 14728
5		T00917	Proteobacteria	<i>Desulfovibrio magneticus</i> RS-1
6		T00969	Proteobacteria	<i>Desulfovibrio salerigens</i> DSM 2638
7		T00452	Proteobacteria	<i>Desulfovibrio vulgaris</i> DP4
8		T00171	Proteobacteria	<i>Desulfovibrio vulgaris</i> Hildenborough
9		T00816	Proteobacteria	<i>Desulfovibrio vulgaris</i> Miyazaki F
10		T02064	Proteobacteria	<i>Desulfovibrio vulgaris</i> RCH1
11		T01382	Proteobacteria	<i>Pseudodesulfovibrio aespocensis</i> ( <i>Desulfovibrio aespocensis</i> ) Aspo-2
12		T04404	Proteobacteria	<i>Pseudodesulfovibrio indicus</i> ( <i>Desulfovibrio indicus</i> ) J2
13		T02481	Proteobacteria	<i>Pseudodesulfovibrio piezophilus</i> ( <i>Desulfovibrio piezophilus</i> ) C1TLV30
1	Cathodic	T01166	Euryarchaeota	<i>Methanobrevibacter ruminantium</i> M1
2		T00539	Euryarchaeota	<i>Methanobrevibacter smithii</i> ATCC 35061
3		T01234	Euryarchaeota	<i>Methanocaldococcus infernus</i> ME
4		T00552	Euryarchaeota	<i>Methanococcus aeolicus</i> Nankai-3
5		T00615	Euryarchaeota	<i>Methanococcus maripaludis</i> C6
6		T00164	Euryarchaeota	<i>Methanococcus maripaludis</i> S2
7		T01584	Euryarchaeota	<i>Methanococcus maripaludis</i> X1
8		T01297	Euryarchaeota	<i>Methanothermobacter marburgensis</i> Marburg
9		T01510	Euryarchaeota	<i>Methanotorris igneus</i> Kol 5

**Table 5.13** Cyanobacteria identified using the anodic protein search.

	Genome ID	Organism	Co-occurrence
1	T02358	<i>Calothrix</i> sp. PCC 7507	Fdh
2	T04195	<i>Fischerella</i> sp. NIES-3754	Fdh
3	T04499	<i>Moorea producens</i> PAL-8-15-08-1	Fdh
4	T05266	<i>Nostoc flagelliforme</i> CCNUN1	Fdh
5	T00713	<i>Nostoc punctiforme</i> PCC 73102	Fdh
6	T04170	<i>Nostoc</i> sp. NIES-3756	Fdh
7	T04905	<i>Nostocales cyanobacterium</i> HT-58-2	Fdh
8	T02367	<i>Pleurocapsa</i> sp. PCC 7327	Fdh
9	T00387	<i>Trichodesmium erythraeum</i> IMS101	Fdh
10	T04059	<i>Anabaena</i> sp. wa102 WA102	Fdh,Flg
11	T00806	<i>Cyanothece</i> sp. PCC 8801	Fdh,Flg
12	T00968	<i>Cyanothece</i> sp. PCC 8802	Fdh,Flg
13	T02341	<i>Anabaena</i> sp. 90	Flg
14	T02608	<i>Arthrospira platensis</i> NIES-39	Flg
15	T03937	<i>Calothrix</i> sp. 336/3	Flg
16	T02362	<i>Nostoc</i> sp. PCC 7107	Flg

levels according to the researcher’s interest, i.e. selecting for a particular phylum, genus, species, etc. For instance, if there was an interest in working with cyanobacteria as the anodic organisms, the genomes belonging to the phylum “Cyanobacteria” could be extracted. Table 5.7 shows that 16 genomes, shown in Table 5.13, were found belonging to this phylum. It can be observed that these genomes have been annotated with Fdh, Flg or a combination thereof. Both approaches can be combined to further select the desired organisms. For example, if the potential of Fdh as an organic electron carrier were to be investigated, organisms 1 through 9 in Table 5.13 could be selected.

The same principle led to 103 Cyanobacteria genomes being identified as potential cathodic organisms. Table 5.14 shows the organism’s genera and the protein co-occurrence observed. Further filtering would be required to obtain a testable number of cathodic organisms. This would depend on the research question and availability of organisms. To mirror previous reasoning, the genome of the genus *Moorea*, *Moorea producens* PAL-8-15-08-1, could be an interesting candidate as it is the only Cyanobacteria listed to potentially be able to achieve direct and indirect electron transfer through c-type cytochromes, Fdh and H<sub>2</sub>. On the other hand, the role of c-type cytochrome could be evaluated using a subset of the genomes annotated only with Cyc.

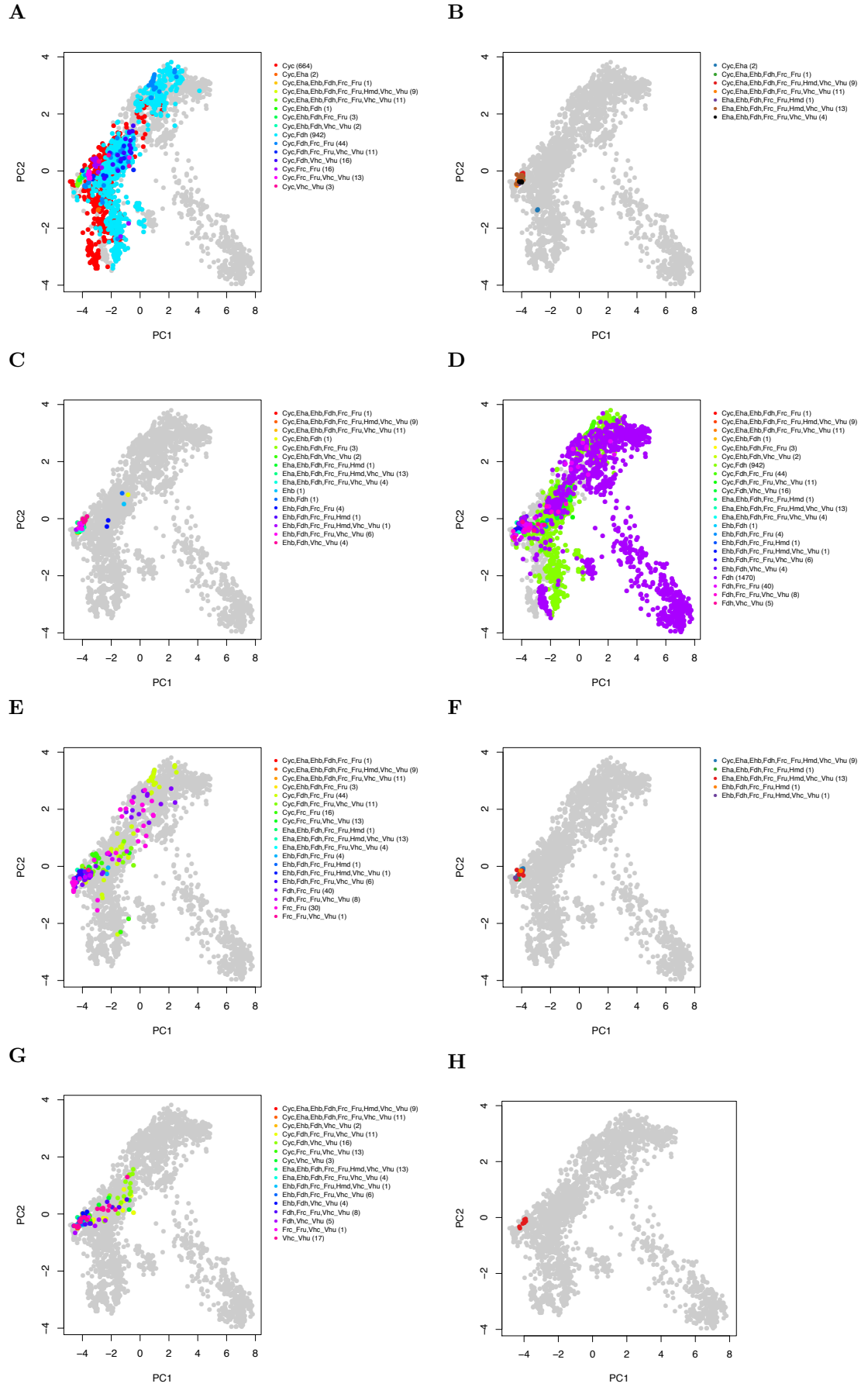
Finally, MetQy can be used as an additional layer to filter the data by using biochemical information as defined by KEGG modules (see Chapter 4, Section 4.2). The genome IDs were used to query all of the modules across the list of anodic and cathodic organisms identified separately. Code 5.2 first generates a matrix for the module completeness fraction (mcf) calculated for every KEGG module (columns) and every genome ID provided (rows). It then performs a principal component analysis (PCA) on the mcf matrix to reduce the dimensionality of the data. Genomes that are similar in their mcf across modules tend to cluster together when the first two principal components (PCs) are plotted in a PC plot.

**Table 5.14** Cyanobacteria identified using the cathodic protein search.

	<b>Genus</b>	<b>Cyc</b>	<b>Cyc, Fdh</b>	<b>Cyc, Fdh, Frc/Fru</b>
1	<i>Acaryochloris</i>	1	0	0
2	<i>Anabaena</i>	2	1	0
3	<i>Arthrospira</i>	1	0	0
4	<i>Calothrix</i>	2	1	0
5	<i>Candidatus</i>	1	0	0
6	<i>Chamaesiphon</i>	1	0	0
7	<i>Chondrocystis</i>	1	0	0
8	<i>Chroococcidiopsis</i>	1	0	0
9	<i>Crinalium</i>	1	0	0
10	<i>Crocospaera</i>	1	0	0
11	<i>Cyanobacterium</i>	3	0	0
12	<i>Cyanobium</i>	2	0	0
13	<i>Cyanothece</i>	4	2	0
14	<i>Cylindrospermum</i>	1	0	0
15	<i>Dactylococcopsis</i>	1	0	0
16	<i>Fischerella</i>	0	1	0
17	<i>Geitlerinema</i>	1	0	0
18	<i>Geminocystis</i>	2	0	0
19	<i>Gloeobacter</i>	2	0	0
20	<i>Gloeocapsa</i>	1	0	0
21	<i>Halomicronema</i>	1	0	0
22	<i>Halothece</i>	1	0	0
23	<i>Leptolyngbya</i>	4	0	0
24	<i>Microcoleus</i>	1	0	0
25	<i>Microcystis</i>	2	0	0
26	<i>Moorea</i>	0	0	1
27	<i>Nostoc</i>	4	3	0
28	<i>Nostocales</i>	0	1	0
29	<i>Oscillatoria</i>	2	0	0
30	<i>Pleurocapsa</i>	0	1	0
31	<i>Prochlorococcus</i>	14	0	0
32	<i>Pseudanabaena</i>	1	0	0
33	<i>Rivularia</i>	1	0	0
34	<i>Stanieria</i>	2	0	0
35	<i>Synechococcus</i>	19	0	0
36	<i>Synechocystis</i>	7	0	0
37	<i>Thermosynechococcus</i>	2	0	0
38	<i>Trichodesmium</i>	0	1	0
39	<i>Trichormus</i>	1	0	0
<b>TOTAL</b>		<b>91</b>	<b>11</b>	<b>1</b>

Figures 5.2 and 5.1 show a PC plot for all the data, overlaying the protein co-occurrence by protein, for the anodic and cathodic protein searched, respectively. It can be observed that the organisms annotated with Hyn (Figure 5.2F) cluster together in an area similar to where genomes annotated with all six anodic proteins cluster (Figure 5.2G). These anodic organisms are listed in Table 5.15. Similarly, the protein co-occurrence including Frc/Fru (Figure 5.1E) cluster together in an area similar to where genomes annotated with all seven cathodic proteins cluster (Figure 5.1H), as well as most protein co-occurrence involving Eha (Figure 5.1B) and Ehb (Figure 5.1C). This led to the identification of 30 different organisms that could be used as cathodic organisms (Table 5.16). These had the protein co-occurrence listed in Table 5.17.

Therefore, these clustering patterns could be used to select biochemically similar anodic and cathodic organisms. Table 5.15 lists the anodic organisms annotated with Hyn, except those



**Figure 5.1** PC plot of all the cathodic organisms identified. The protein co-occurrence was overlaid by protein. Co-occurrence including **A** Cyc, **B** Eha, **C** Ehb, **D** Fdh, **E** Frc/Fru, **F** Hmd, **G** Vhc/Vhu, **H** Co-occurrence of all seven proteins.





**Table 5.15** Genomes found to be annotated with Hyn.

	Genome ID	Phylum	Organism	Co-occurrence
1	T01284	Proteobacteria	<i>Desulfarculus baarsii</i> DSM 2075	Coo,Fdh,Flg,Hyn
2	T00813	Proteobacteria	<i>Desulfatibacillum alkenivorans</i> AK-01	Coo,Fdh,Flg,Hyn
3	T00861	Proteobacteria	<i>Desulfobacterium autotrophicum</i> HRM2	Coo,Fdh,Flg,Hyn
4	T02473	Proteobacteria	<i>Desulfocapsa sulfexigens</i> DSM 10523	Coo,Fdh,Flg,Hyn
5	T00989	Proteobacteria	<i>Desulfohalobium retbaense</i> DSM 5692	Ccm,Coo,Fdh,Flg,Hyn
6	T00963	Proteobacteria	<i>Desulfomicrobium baculatum</i> DSM 4028	Ccm,Coo,Fdh,Flg,Hyn
7	T02147	Proteobacteria	<i>Desulfomonile tiedjei</i> DSM 6799	Coo,Fdh,Hyd,Hyn
8	T01662	Proteobacteria	<i>Desulfovibrio africanus</i> Walvis Bay	Ccm,Coo,Fdh,Flg,Hyn
9	T01709	Proteobacteria	<i>Desulfovibrio desulfuricans</i> ND132	Ccm,Coo,Fdh,Flg,Hyn
10	T02851	Proteobacteria	<i>Desulfovibrio gigas</i> DSM 1382	Ccm,Coo,Fdh,Flg,Hyn
11	T04638	Proteobacteria	<i>Desulfovibrio piger</i> FI11049	Ccm,Coo,Fdh,Hyd,Hyn
12	T00521	Proteobacteria	<i>Geobacter uraniireducens</i> Rf4	Coo,Fdh,Flg,Hyn
13	T01456	Proteobacteria	<i>Hipaea maritima</i> DSM 10411	Coo,Flg,Hyn
14	T00418	Proteobacteria	<i>Syntrophobacter fumaroxidans</i> MPOB	Coo,Fdh,Hyd,Hyn
15	T01547	Thermodesulfobacteria	<i>Thermodesulfobacterium geofontis</i> OPF15	Coo,Fdh,Flg,Hyn

organisms annotated with all six anodic proteins listed in Table 5.12. The genomes annotated with the three proteins identified in the cathodic search (Eha, Ehb and Frc/Fru), excluding the genomes annotated with all seven, are listed in Table 5.16. This would result in  $15 \times 30 = 450$  combinations of anodic and cathodic organisms. Further information, such as the potential to have a specific biochemical process (KEGG module) or a phylogenetic level would be needed to

**Table 5.16** Genomes found to be annotated with Eha, Ehb and Frc/Fru.

	Genome ID	Organism	Co-occurrence
1	T04503	<i>Methanobacterium curvum</i> Buetzberg	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
2	T03396	<i>Methanobacterium formicicum</i> BRM9	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
3	T03743	<i>Methanobacterium formicicum</i> DSM1535	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
4	T01437	<i>Methanobacterium lacus</i> ( <i>Methanobacterium</i> sp. AL-21)	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
5	T01509	<i>Methanobacterium paludis</i> ( <i>Methanobacterium</i> sp. SWAN-1)	Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
6	T02957	<i>Methanobacterium</i> sp. MB1	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
7	T04203	<i>Methanobrevibacter millerae</i> SM9	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
8	T04385	<i>Methanobrevibacter olleyae</i> YLM1	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
9	T02687	<i>Methanobrevibacter</i> sp. AbM4	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
10	T04279	<i>Methanobrevibacter</i> sp. YE315	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
11	T03253	<i>Methanocaldococcus bathoardescens</i> JH146	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
12	T00976	<i>Methanocaldococcus fervens</i> AG86	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
13	T00003	<i>Methanocaldococcus jannaschii</i> DSM 2661	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
14	T01175	<i>Methanocaldococcus</i> sp. FS406-22	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
15	T01105	<i>Methanocaldococcus vulcanius</i> M7	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
16	T00491	<i>Methanococcus maripaludis</i> C5	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
17	T00553	<i>Methanococcus maripaludis</i> C7	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
18	T00554	<i>Methanococcus vannieli</i> SB	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
19	T01251	<i>Methanococcus voltae</i> A3	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
20	T00472	<i>Methanocorpusculum labreanum</i> Z	Eha,Ehb,Fdh,Frc/Fru,Hmd
21	T02189	<i>Methanoculleus bourgensis</i> MS2T	Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
22	T00477	<i>Methanoculleus marisnigri</i> JR1	Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
23	T04314	<i>Methanoculleus</i> sp. MAB1	Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
24	T01312	<i>Methanolacinia petrolearia</i> DSM 11571	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
25	T00078	<i>Methanopyrus kandleri</i> AV19	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
26	T00325	<i>Methanospirillum hungatei</i> JF-1	Cyc,Eha,Ehb,Fdh,Frc/Fru
27	T04009	<i>Methanothermobacter</i> sp. CaT2	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
28	T00009	<i>Methanothermobacter thermautotrophicus</i> Delta H	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu
29	T04587	<i>Methanothermobacter wolfeii</i> SIV6	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu
30	T01534	<i>Methanothermococcus okinawensis</i> IH1	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu

**Table 5.17** Co-occurrence of cathodic genomes annotated with Eha, Ehb and Frc/Fru.

	Co-occurrence	Freq
1	Cyc,Eha,Ehb,Fdh,Frc/Fru	1
2	Eha,Ehb,Fdh,Frc/Fru,Hmd	1
3	Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu	4
4	Cyc,Eha,Ehb,Fdh,Frc/Fru,Vhc/Vhu	11
5	Eha,Ehb,Fdh,Frc/Fru,Hmd,Vhc/Vhu	13

reduce the number of pairs to suit experimental capacity.

### 5.2.5 Generation of testable hypotheses based on literature findings

MetQy can also be used to generate testable hypothesis of possible organism pairings through electrochemical means based on previous coculture studies (see Code 5.3). For example, Hill et al. (2017) developed a platform to coculture the methanotrophic bacterium *Methylobacterium alcaliphilum* 20z and the cyanobacterium *Synechococcus* sp. PCC 7002, in order to utilise the greenhouse gasses CH<sub>4</sub> and CO<sub>2</sub> to produce microbial biomass. By searching the organism names across the anodic and cathodic genomes identified in this work, *Methylobacterium alcaliphilum* (genome T01649) was found among the identified anodic organisms. Additionally, *Synechococcus* sp. PCC 7002 (genome T00664) was identified as a cathodic organism. Both genomes were annotated with c-type cytochrome biogenesis proteins (Ccm and Cyc for the methanotroph and the cyanobacterium, respectively), suggesting that both could carry out indirect electron transfer as proposed by Croese et al. (2011). Moreover, *Methylobacterium alcaliphilum* was annotated with Fdh and Flg, again suggesting the possibility of indirect electron transfer, as well as the possibility of direct long-range electron transfer by means of a flagellum. Therefore, MetQy can easily facilitate the data mining of literature-based information across genes for over 5,000 genomes, allowing the generation of testable hypothesis.

## 5.3 Discussion

The aim of this chapter was to demonstrate how MetQy can be used to inform electrochemical experiments to further investigate the “syntrophy over wires” hypothesis presented in this work. The proposed mechanisms by which Dv and Mm perform EET and, thus, interact with electrodes was used to identify genomes annotated with genes that would potentially enable the same type of EET mechanisms. These included six and seven proteins for Dv and Mm, respectively, which were successfully mapped to KEGG orthologs (K numbers) (see Tables 5.1 and 5.2). This enabled the identification of 3,504 and 3,345 genomes that could also interact with electrodes with some or all of the mechanisms proposed for Dv and Mm, respectively, as summarised in Tables 5.3 and 5.4.

The analysis performed in this chapter was liberal as a single protein subunit was required for the assumption that the entire protein was annotated in the genomes identified. This was done to account for gene miss-annotations in either the genome or in the mapping between the genes and the K numbers. It is reasonable to assume that the results could change significantly if a more strict approach were followed. In order to implement a more stringent approach, a manual curation would be required to define the K numbers conforming every protein.

The results of the analysis were successfully validated by confirming that Dv and Mm were cross-identified as expected, since they were both annotated with *fdhD*. Mm was also retrieved by being annotated with the anaerobic carbon-monoxide dehydrogenase iron sulphur subunit (CooF, K00196). This was not expected as there has been no report of Coo being annotated in Mm's genome in the literature to the author's knowledge. However, the UniProt database (Bateman et al., 2017) indicates that an entry has been made for *Methanococcus maripaludis* (*Methanococcus deltae*) with the predicted gene *porF*, which has the protein name: "iron-sulphur protein" and has been given the synonym "*cooF*". Since the annotation has not been reviewed it is not possible at this time to determine whether Coo is correctly annotated in Mm's genome or not. Further research is required to validate this K number assignment to Mm's genome.

Furthermore, the identification of known electroactive microorganisms was used as a means to validate the method. The same 11 and 26 *Geobacter* and *Shewanella* genomes, respectively, were identified with the anodic and cathodic protein searches (Table 5.10 and Table 5.11). A literature search was carried out to determine which genomes have been found capable of EET and the relevant studies are listed in the latter table. Most *Geobacter* genomes listed have been used in EET applications, while efforts have mostly focused on using *S. oneidensis* MR-1 rather than other *Shewanella* species. Further research is needed to determine whether the genomes with no studies identified are capable of EET and, if so, by which mechanism(s).

During this particular analysis, over 3,000 genomes were retrieved as potential anodic and cathodic organisms, rendering an impractical number in terms of experimental validation. The selection of testable organisms should be defined by the particular experimental question at hand. Here, different approaches were pursued assuming different research interests as examples by using the information retrieved as part of this data mining exercise using MetQy. These led to multiple subsets of organism pairs to be obtained that could be used to inform experiments. The co-occurrence of proteins (Tables 5.5 and 5.6) and the taxonomic information contained within KEGG genome, as well as the relationship between these (Tables 5.7 and 5.8) were used to filter the organisms identified.

The first approach consisted in retrieving anodic and cathodic organisms annotated with all Dv's and Mm's proteins, respectively. In this manner, the anodic organisms were reduced to 13

genomes, all from the genera *Desulfovibrio* and *Pseudodesulfovibrio* and including Dv, while the cathodic organisms were reduced to 9 genomes from the genera *Methanobrevibacter*, *Methanocaldococcus*, *Methanococcus*, *Methanothermobacter*, *Methanotorris*, including Mm. These resulted in 116 possible pairs of organisms that could be tested under the “syntrophy over wires” hypothesis.

The second approach assumed an interest with working with members of a particular phylum, Cyanobacteria. 16 genomes were identified as potential anodic organisms (Table 5.13). However, 103 genomes were identified as potential cathodic organisms (Table 5.14), not limiting the number of pairings down to a testable set. It was proposed that either the potential mechanism by which EET would take place (indirect or a combination of direct and indirect) or the taxonomy could be used to further select the cathodic organism.

Finally, it was proposed that the information on the biochemical processes of the genomes could also be used as a filtering method. This was achieved by using `query_genomes_to_modules`, a MetQy function, to obtain the module completeness fraction (mcf) matrix for the KEGG modules across the identified anodic and cathodic organism. PCA was then performed on the mcf matrix to reduce the dimensionality of the data and to highlight similarities between genomes in the form of clusters when plotting the first two PCs. Clusters could indicate genomes that share similar biochemical processes profiles. The protein co-occurrence was overlaid to further highlight similarities between genomes annotated with the different proteins and the genomes annotated with all the proteins. Therefore, the clusters of the genomes annotated with all the anodic or cathodic proteins were used to identify protein co-occurrence combinations leading to similar biochemical profiles. It was determined that genomes annotated with Hyn (in any protein co-occurrence) shared a profile similar to those annotated with Dv’s proteins, which led to the identification of 15 potential anodic organisms (Table 5.15). On the other hand, genomes annotated with Eha, Ehb or Frc/Fru had profiles similar to those organisms annotated with Mm’s proteins. Since the combination of genomes annotated with any of these three proteins led to the identification of over 100 genomes, protein co-occurrence was limited to the presence of minimum those three proteins, resulting in 30 potential cathodic organisms (Table 5.16). Interestingly, the protein co-occurrence for these genomes (Table 5.17) showed that there are more genomes that have six out of seven proteins, rather than fewer. This would suggest that the annotation motif Eha, Ehb, Fdh, Frc/Fru, Vhc/Vhu (row 3 in Table 5.17) is frequent and often found with additional annotations.

The generation of testable hypotheses based on literature findings was also demonstrated by determining whether the members of a published coculture (Hill et al., 2017) could be capable of EET, therefore proposing a new means of interaction between the two organisms. Preliminary experiments could be conducted to determine if the organisms are capable of EET. If those were successful, an experiment could be designed based on this proposed interaction to test the hypoth-

esis growing *Methylobacterium alcaliphilum* 20z and *Synechococcus* sp. PCC 7002 on separate electrodes.

On another note, **MetQy**'s ease of use was demonstrated by being able to extract information on over 5,000 genomes with a few lines of code given a table containing K numbers corresponding to proteins of interest. The 'traditional' means by which a similar data extraction could be to perform a protein Basic Local Alignment Search Tool (BLASTP; Altschul et al., 1997). This process would be time-consuming as it would involve either the manual search through NCBI's Web BLAST interface<sup>1</sup> or the downloading of BLAST+ and then generating custom-made bash (command-line) scripts to automate the search across multiple genes. The advantage of using BLASTP would be that the data is up-to-date, while **MetQy** relies on in-built data. This was highlighted when K22516 (*fdhA*, formate dehydrogenase alpha subunit) was used to retrieve genomes. However, this limitation can be addressed by having FTP access to KEGG as **MetQy**'s functions were designed to take up-to-date information from which the data mining is performed.

As with any bioinformatics tool, the accuracy of **MetQy**'s functions is determined by the quality of the data stored within KEGG. Redundancy was observed between the mapping of RefSeq or GenBank genes with K numbers. For instance, the genes for both Frc and Fru, as well as those for Vhc and Vhu, were found to be mapped to the same set of K numbers. This was taken into account during the analysis and Frc and Fru (and Vhc and Vhu) were considered as a single protein. As with the annotation of CooF in Mm's genome, an inconsistency was observed during the mapping between RefSeq genes and K numbers where the genes were annotated for c-type cytochrome biogenesis genes, but the K numbers referred to heme transport proteins. Building databases such as the KEGG ortholog used here requires on-going work and curation. Users of tools using KEGG data, such as **MetQy** should keep these limitations in mind. The last inconsistency has been reported to the KEGG maintainer and the author hopes the mapping will be re-evaluated. This also highlights the need for experimental validation of data retrieved from KEGG.

The work presented in this chapter also reflected the flexibility of analysis that is possible through the use of **MetQy**. The anodic and cathodic protein searches resulted in thousands of genomes that could be used to test the "syntrophy over wires" hypothesis by substituting Dv and Mm. Due to the large number of genomes, several means of filtering were proposed, in order to reduce the organisms to a testable number, feasible due to the information retrieved through the use of **MetQy** functions. These included filtering by the protein co-occurrence, by phylogenetic level or by the similarity of the genomes' biochemical processes (defined by KEGG modules) to Dv and Mm.

---

<sup>1</sup><https://blast.ncbi.nlm.nih.gov/Blast.cgi>

## 5.4 Methods

This section includes the code used for the analysis described in this chapter. The code presented here is meant to be run within the R environment. The code snippets include the scripts and functions used. Comments begin with a hash symbol (`#` and are in green), functions and keywords are highlighted in blue and text or strings are in magenta.

### 5.4.1 Identify genomes from K numbers

The following code (Code 5.1) states the steps followed for every set of K numbers corresponding to a protein. The protein to K number mapping can be observed in Tables 5.1 and 5.2. Note that these steps were carried out for each protein independently. The resulting organisms found were compared as presented in the Results section. The code for the manipulation of the extracted data is not shown.

The `genome_reference_table` object used in this code section is the formatted KEGG genome database, obtained by parsing the KEGG file using `parseKEGG_genome`. The generation of this object requires FTP access to KEGG. However, the only information extracted from the object `genome_reference_table` was the genome ID, organism name and the taxonomic information. Appendix Code F.1 demonstrates a ‘hack’ to be able to obtain this information using `MetQy` functions without requiring FTP access.

#### Code 5.1 Identifying genomes from K numbers

```
1 identification_genomes_from_Knumbers <- function(info_table , org ,
2                                                  genome_reference_table){
3 # EXPECTED INPUT
4 # info_table (data frame):
5 #           Enzyme K_number
6 #           1   Coo   K00196
7 #           2   Coo   K00198
8 #           3   Hyn   K18008
9 #           4   Hyn   K00437
10 #           5   Hyd   K00533
11 #           ...
12 #
13 # org (string):
14 #           "Dy"
15 # genome_reference_table (data frame)
16 # NOTE: see main text for a hack to get genome_reference_table object
17
18 # LOAD LIBRARIES AND DATA ———
19 library(MetQy)
```

```

20
21 # PREPARE OUTPUT STORAGE AND RECORD ORG USED AS BASE (source of enzymes/proteins)
22 output <- vector("list",1)
23 names(output) <- org
24
25 # GET LIST OF ENZYMES OR PROTEINS
26 enzymes <- sort(unique(info_table$Enzyme))
27
28 # STORE SUMMARY
29 DATA_summary_table <- data.frame("Enzyme" = enzymes ,
30                                   "Gene_ID" = '' ,
31                                   "K_numbers" = '' ,
32                                   "No of genomes matched" = 0 ,
33                                   "No of unique organisms" = 0 ,
34                                   stringsAsFactors = F)
35
36 # RETRIEVE GENOMES FOR EACH ENZYME AND STORE THEIR GENOME INFO
37 pivot_table <- NULL # prepare store
38 genomes_enzyme <- vector("list",length(enzymes))
39 for(e in 1:length(enzymes)){
40   # GET numbers from this enzyme
41   index <- which(info_table$Enzyme==enzymes[e])
42   K_numbers <- unique(info_table$K_number[index])
43
44   # Get genomes that have been annotated with those K numbers using the MetQy
   package function
45   genome_gene_table <- query_genes_to_genomes(genes = K_numbers)
46   # genome_gene_table[1:3,1:3]
47   #           T00003 T00009 T00011
48   # K14126      1      1      1
49   # K14127      0      1      1
50   # K14128      1      1      1
51
52   genomes_enzyme[[e]]$genome_gene_table <- genome_gene_table
53   genomes <- colnames(genome_gene_table)
54
55   # Use the genome IDs to get information stored in the KEGG genome database in
   object genome_reference_table
56   matched_genomes <- genome_reference_table[match(genomes ,
57   genome_reference_table$ID) ,]
58   matched_genomes <- matched_genomes[order(matched_genomes$ORGANISM) ,]
59
   # Save information to the summary table

```



```

60 DATA_summary_table$Gene_ID[e] <- paste(info_table$Gene_ID[index], collapse = ",
    ")
61 DATA_summary_table$K_numbers[e] <- paste(K_numbers, collapse = ", ")
62 DATA_summary_table$No. of . genomes . matched[e] <- ncol(genome_gene_table)
63 DATA_summary_table$No. of . unique . organisms[e] <- length(unique(
    matched_genomes$ORGANISM))
64
65 # Create a table with the name of the enzyme/protein, the genome IDs,
    the organisms' names and their taxonomy
66 tmp_store <- cbind(rep(enzymes[e], nrow(matched_genomes)), matched_genomes$ID,
    matched_genomes$ORGANISM, matched_genomes$TAXONOMY)
67 pivot_table <- rbind(pivot_table, tmp_store)
68 }
69 pivot_table <- data.frame(pivot_table, stringsAsFactors = F)
70 # Name the variables
71 names(pivot_table) <- c("MATCHED.ENZYME", "ID", "ORGANISM", "TAXONOMY")
72 # pivot_table[1,]
73 # MATCHED.ENZYME ID ORGANISM TAXONOMY
74 # 1 Ccm T04735 Acetobacter aceti TMW2.1153 TAX:435;LINEAGE Bact [...]
    Acetobacter
75
76 # Get the total number of unique genome IDs
77 total_genomes <- length(unique(pivot_table$ID))
78 # Get how many times each genome ID was identified
79 genomes_matched_enzymes <- table(pivot_table$ID)
80 # Count how many times the genomes were identified 1, 2, 3, ... times
81 occurrence <- table(genomes_matched_enzymes)
82 # print(occurrence) -----
83 # 1 2 3 | 4 | 5 6 —> no. enzymes annotated across genomes
84 # 1121 1260 1058 | 46 | 6 13 —> counts
85 # ----- : e.g. 46 genomes were annotated with 4 enzymes
86
87 n_matches <- names(occurrence) # 1 2 3 4 5 6 above
88 n_matches <- as.numeric(n_matches)
89
90 # Look at which genomes were annotated with which enzymes
91 matched <- vector("list", length(occurrence))
92 names(matched) <- paste("matched_", n_matches, sep="")
93 genomes_matched <- vector("list", length(occurrence))
94 names(genomes_matched) <- paste("genomes_matched_", n_matches, sep="")
95 for( n in 1:length(occurrence)){
96     matched[[n]] <- names(genomes_matched_enzymes)[which(genomes_matched_enzymes==
        n_matches[n])]

```

```

97     genomes_matched[[n]] <- genome_reference_table[match(matched[[n]],
genome_reference_table$ID),]
98     genomes_matched[[n]] <- genomes_matched[[n]][order(genomes_matched[[n]]
$ORGANISM),]
99 }
100
101 # Get taxonomic information using MetQy function
102 matched_genomes_tax <- vector("list",length(ocurrence))
103 names(matched_genomes_tax) <- paste("matched_genomes_",n_matches,sep="")
104 for( n in 1:length(ocurrence)){
105     matched_genomes_tax[[n]] <- parseKEGG_process_KEGG_taxonomy(genomes_matched[[n
]])
106     genomes_matched[[n]]$KINGDOM <- matched_genomes_tax[[n]]$KINGDOM
107     genomes_matched[[n]]$PHYLUM <- matched_genomes_tax[[n]]$PHYLUM
108     genomes_matched[[n]]$GENUS <- matched_genomes_tax[[n]]$GENUS
109 }
110
111 # Summarise info
112 matched_data <- vector("list",length(ocurrence))
113 names(matched_data) <- paste("matched_data_",n_matches,sep="")
114 for(n in 1:length(ocurrence)){
115     matched_data[[n]] <- data.frame("ID" = genomes_matched[[n]]$ID,
116                                     "PHYLUM" = genomes_matched[[n]]$PHYLUM,
117                                     "ORGANISM" = genomes_matched[[n]]$ORGANISM,
118                                     stringsAsFactors = F)
119     for(e in 1:length(enzymes)){
120         matched_data[[n]] <- cbind(matched_data[[n]], as.character(as.numeric(!is.na(
match(genomes_matched[[n]]$ID, names(genomes_enzyme[[e]]$genome_gene_table))))))
121         names(matched_data[[n]])[ncol(matched_data[[n]])] <- enzymes[e]
122     }
123     matched_data[[n]] <- matched_data[[n]][order(matched_data[[n]]$ORGANISM),]
124 }
125
126 # Collect all of the genomes matched
127 matched_data_flat <- NULL
128 for(n in 1:length(ocurrence)){
129     matched_data_flat <- rbind(matched_data_flat,matched_data[[n]])
130 }
131
132 # Set correct data type
133 for(C in 1:ncol(matched_data_flat)){
134     matched_data_flat[,C] <- as.character(matched_data_flat[,C])
135 }

```

```

136
137 # FIND COMBINATION OF ENZYMES IDENTIFIED
138 for(e in 1:length(enzymes)){
139   which_col <- which(names(matched_data_flat)==enzymes[e])
140   matched_data_flat[which(matched_data_flat[,which_col]=="1"),which_col] <-
141     enzymes[e]
142 }
143
144 matched_data_flat <- matched_data_flat[order(matched_data_flat$ID),]
145 enzymes_matched <- apply(matched_data_flat[,4:ncol(matched_data_flat)], 1, paste,
146   collapse = ", ")
147 enzymes_matched <- gsub("[,]\\s0" , "" , enzymes_matched)
148 enzymes_matched <- gsub("0[,]\\s" , "" , enzymes_matched)
149
150 matched_data_flat <- matched_data_flat[order(enzymes_matched),]
151 enzymes_matched <- enzymes_matched[order(enzymes_matched)]
152
153 matched_data_flat$COOCCURRENCE <- enzymes_matched
154
155 # Store in output
156 output[[1]]$DATA_summary_table <- DATA_summary_table
157 output[[1]]$pivot_table <- pivot_table
158 output[[1]]$ocurrence <- ocurrence
159 output[[1]]$matched <- matched
160 output[[1]]$matched_genomes_tax <- matched_genomes_tax
161 output[[1]]$genomes_matched <- genomes_matched
162 output[[1]]$matched_data <- matched_data
163 output[[1]]$matched_data_flat <- matched_data_flat
164 output[[1]]$enzymes_matched <- enzymes_matched
165 return(output)
166 }

```

### 5.4.2 Selecting genome subsets using biochemical process information

Code used to obtain the biochemical processes that each genome can carry out as defined by the module completeness fraction (mcf) for each KEGG module. PCA was carried out to reduce the dimensionality of the data and identify clustering of similar genomes.

**Code 5.2** Use of MetQy to determine organism subset

```

1 organism_subset_by_post_analysis <-function(matched_data_flat ,enzymes_matched ,org){
2 # NOTE: input generated with identification_genomes_from_Knumbers( ).
3
4 # LOAD LIBRARIES ———

```

```

5 library(MetQy)
6
7 # Get all the module completeness fraction (mcf) for all KEGG modules across the
  genomes identified
8 mcf_info <- query_genomes_to_modules(GENOMEINFO = matched_data_flat$ID)
9 mcf <- mcf_info$MATRIX
10
11 # Visualise mcf matrix
12 p <- plot_heatmap(mcf, Filename = paste("fig/",org,"_heatmap-all.png",sep=""))
13 pca <- prcomp(mcf)
14
15 p <- plot_scatter_byFactors(pca$x, FACTOR = list("enzymes" = enzymes_matched),
  Filename = paste("fig/",org,"_pc-plot.pdf",sep=""))
16
17 # fdh is everywhere ——
18 fdh_index <- which(enzymes_matched=="fdh")
19 # enzymes_matched_rm_fdh <- enzymes_matched[-fdh_index]
20
21 # OVERLAY ALL GROUPS EXPECT FDH
22 enzymes_fdh_null <- enzymes_matched
23 enzymes_fdh_null[fdh_index] <- NA
24
25 p <- plot_scatter_byFactors(pca$x, FACTOR = list("enzymes" = enzymes_fdh_null),
  Filename = paste("fig/",org,"_pc-plot-fdh-grey.pdf",sep=""))
26
27 # OVERLAY GROUPS CONTAINING ENZYME (except fdh by itself) ——
28 enzyme_factors <- vector("list", length(enzymes))
29 names(enzyme_factors) <- enzymes
30 for(e in 1:length(enzymes)){
31   this_factor <- rep(NA,length(enzymes_matched))
32   index <- grep(enzymes[e],enzymes_matched)
33   this_factor[index] <- enzymes_matched[index]
34   enzyme_factors[[e]] <- this_factor
35 }
36 p <- plot_scatter_byFactors(pca$x, FACTOR = enzyme_factors, Filename = paste("fig
  /",org,"_pc-plot-enzyme-groups.pdf",sep=""))
37
38 # OVERLAY GENOMES THAT HAVE THE MOST ENZYMES ——
39 enzymes_all <- enzymes_matched
40 enzymes_all[which(nchar(enzymes_matched)<max(nchar(enzymes_matched)))] <- NA
41
42 p <- plot_scatter_byFactors(pca$x, FACTOR = list("enzymes-all" = enzymes_all),
  Filename = paste("fig/",org,"_pc-plot.pdf",sep=""), Width = 5)

```

```

43
44 # Store output
45 output <- vector("list",1)
46 names(output) <- org
47 output[[1]]$mcf <- mcf
48 output[[1]]$enzymes_fdh_null <- enzymes_fdh_null
49 output[[1]]$enzyme_factors <- enzyme_factors
50 output[[1]]$enzymes_all <- enzymes_all
51 return(output)
52 }

```

### 5.4.3 Generation of testable hypotheses based on literature findings

Below is the code used to identify the organisms used by Hill et al. (2017) as anodic and cathodic organisms.

**Code 5.3** Using MetQy to check the possible pairing of a known coculture in the context of the “syntrophy over wires” hypothesis

```

1 #' Data obtained with identification_genomes_from_Knumbers():
2 #'   REQUIRED OBJECTS:
3 #'   anodic_organisms, cathodic_organisms
4
5 head(anodic_organisms)
6 #      ID          PHYLUM          ORGANISM COOCCURRENCE
7 # 1  T00018      Proteobacteria Rickettsia prowazekii Madrid E      Ccm
8 # 2  T00023      Crenarchaeota      Aeropyrum pernix K1      Ccm
9 # 3  T00025 Deinococcus-Thermus    Deinococcus radiodurans R1      Ccm
10
11 ## Look to see if Methylomicrobium alcaliphilum and Synechococcus sp. PCC 7002
12 #   were identified as anodic and cathodic organisms, respectively.
13 anodic_organisms[ grep("Methylomicrobium alcaliphilum", anodic_organisms$ORGANISM),]
14 #      ID          PHYLUM          ORGANISM Ccm COOCCURRENCE
15 # 2853 T01649 Proteobacteria Methylomicrobium alcaliphilum Ccm Ccm, Fdh, Flg
16 cathodic_organisms[ grep("Synechococcus sp. PCC 7002", cathodic_organisms$ORGANISM),]
17 #      ID          PHYLUM          ORGANISM COOCCURRENCE
18 # 1920 T00664 Cyanobacteria Synechococcus sp. PCC 7002      Cyc

```

## Chapter 6

# Conclusions

The work presented in this thesis was in the context of thermodynamic considerations in the study of metabolism. The focus was on metabolic interactions of a syntrophic coculture through IHT and its possible implementation as a BES. To address these questions, a suitable experimental electrochemistry platform had to first be developed to then carry out biological experiments. Furthermore, to enable systematic analysis of the diversity of EET and to enable the future investigation of the energetics of metabolic redox processes, a computational tool, *MetQy*, was developed.

### 6.1 Development of an electrochemical platform

The first aim within this PhD work was to develop an experimental platform that would enable the investigation of the “syntrophy over wires” hypothesis. The platform required to maintain strict anaerobic conditions for at least 3 weeks and to host a four-electrode system, unlike the standard BES set-up requiring an anode (or working electrode), cathode (or counter electrode) and reference electrode. To overcome some of the practical limitations, such as lack of standardised electrochemical cells and anaerobic conditions maintenance, we collaborated with Dr. James Stratford of (Asally’s group, SLS, Warwick) and Daniel Carlotta-Jones (WMG, Warwick) to design a suitable electrochemical cell. Additionally, Dr. Stratford contributed to the notion of containing the electrochemical cells in an effort to maintain anaerobic conditions and to the development of a protocol to produce low-cost Ag/AgCl electrodes that could be used within the platform.

The aim was achieved with the development of a purpose-built, open-source, generic platform (Figure 2.1). Its modularity and use of a multiplexer (MUX) enabled me to perform the complete experimental design, requiring 12 electrochemical cells, simultaneously. The platform offers flexibility in the choice of electrode material and reactor mode. The electrochemical cell design could be easily modified to contain threaded ports on the sides to allow gas sparging or the

electrochemical cell cultures to be operated in a continuous mode (see Chapter 2 and 3's Discussion sections). The incorporation of turbidity monitoring of the liquid phase based on a side project done during this PhD (Sasidharan et al., 2018) was proposed (see Chapter 2's Discussion section).

## 6.2 Investigation of the “syntrophy over wire” hypothesis

The investigation of the “syntrophy over wire” hypothesis was initiated with the combination of multiple electrochemical and chemical analytic methods. The absence of growth in both the electrochemical cells and control cultures, except Dv's monoculture, suggested that temperature control is indispensable as the microorganisms showed temperature-related growth inhibition. Once this is tackled, a complete study of this interaction can be carried out.

Most BES studies are a biotechnological application designed for the production of either power (e.g. Liu et al., 2005; Rabaey et al., 2005; Yong et al., 2014) or value-added chemicals (e.g. Awate et al., 2017; Jiang et al., 2014; Roy et al., 2015; Rozendal et al., 2008; Schievano et al., 2016; Wang et al., 2018) sometimes coupled with bioremediation or biorecovery (biomining) applications (e.g. Beckmann et al., 2016; Das et al., 2018; Gajda et al., 2014; Gude, 2015; Gajda et al., 2015, 2018; Gimkiewicz and Harnisch, 2013; Huang et al., 2014; Ieropoulos et al., 2016; Merino Jimenez et al., 2017; Min and Logan, 2004; Nanchaiah et al., 2016; Yu, 2015). Instead, the focus in this work has been on quantifying (accurately) the electron transfer between two organisms in a type of system that has not been reported to the author's knowledge. This has presented several challenges that are not normally encountered in BES studies. For example, the aims meant that minute currents had to be measured, avoiding any external noises. To properly test the hypothesis, several control cases had to be included, which are normally ignored and not included in BES studies, leading to a larger design of experiment that is typical. Finally, the BES had to maintain strict anaerobic conditions due to fragility of microorganisms involved. Therefore, this work can pave the road for future studies by addressing these issues and providing a first experimental attempt towards understanding a possible syntrophy over wires. Achieving this would provide a direct measurement of metabolic rates (and ability to interact with electrodes).

The work presented here sought to correlate electrons measured with cell metabolism through changes in metabolite concentrations since the number of cells could not be estimated. Therefore, a mass balance could be achieved for the entire system, featuring as a strength of this approach. A complete mass balance in the system could provide better insight on how microorganisms interact with electrodes and this work could pave the way for future studies geared towards characterising and better understanding these interactions.

Should the hypothesis prove to hold, this syntrophic coculture “over wires” could potentially help address some of the open questions listed in the introduction, such as how does cell-to-cell

electron transfer occur by decoupling the syntrophic partners and providing a tool to characterise the microorganisms. Additionally, it has been previously established that redox reactions are responsible for energy harvesting and the overall potential energy that can be extracted is limited by the available TEA with the most oxidative (positive) potential. Therefore, an organisms' metabolism could, in theory, be modulated with an electronic interface.

### 6.3 Development of a computational tool to analyse the relationship between genetic information and biological function

The final aim of this PhD work was to develop a computational tool to enable the automated, large-scale analysis of annotated genomes with metabolic information for the discovery and design of electronic microbial interactions. This aim was met through the development of an R package, **MetQy** (Chapter 4), that facilitates data mining of genomic and metabolic information from KEGG (Kanehisa et al., 2017), one of the largest collections of computerised biological knowledge. Chapter 5 demonstrated how **MetQy** can be successfully used to identify syntrophic pairs that could be potentially used to substitute Dv and Mm when testing the “syntrophy over wires” hypothesis. This was achieved by mapping the key proteins involved in EET mechanisms to KEGG orthologs (K numbers) and using **MetQy**'s in-built KEGG data to identify genomes with a similar genetic capacity. Furthermore, **MetQy**'s functions enabled a systematic filtering of those genomes identified based on the genome's taxonomy, biochemical processes or protein annotations. This method could be easily adapted to identify other types of interactions and could provide further insights into the taxonomic and mechanistic diversity of EET.

As previously discussed, the automated analyses of metabolic functions in the context of biogeochemistry processes, such as  $\text{NO}^{-3}/\text{NO}^{-2}$  respiration or methanogenesis (Falkowski et al., 2008), would only be feasible by the implementation of thermodynamic information, as examined in Section 1.2 and represented in Figures 1.2 and 1.3 (and Tables A.1 and A.2). However, this task has proven challenging for several reasons, namely data availability, standardisation and the impact of the environmental conditions on the thermodynamics. Future efforts should definitely be spent on the extension of **MetQy** to incorporate thermodynamics in order to be better able to systematically address questions related to biogeochemistry processes and EET mechanisms, as well as their ecological uses and implications.

### 6.4 Achievements and possible future developments

The work presented in this thesis was focused around the development of experimental and computational tools that could be implemented in multiple ways to help the scientific community.



New-comers to the field of microbial electrochemistry could use the platform described in Chapter 2 to dive into experiments rather than having to spend time designing a custom system, thanks to the detailed preparation and assembly protocol presented. Furthermore, the electrochemical experimental platform could easily be used to implement new types of BES or different configurations thereof. The control experiments necessary to investigate the “syntrophy over wires” hypothesis (Chapter 3) would also increase our knowledge of Dv’s and Mm’s EET mechanisms. Additionally, a new type of metabolic interaction (an electric connection replacing a molecular interaction) was proposed through this hypothesis that could provide useful insight into BES and EET mechanisms, regardless of whether the hypothesis is proven to be true.

**MetQy**, described in Chapter 4, could be used in a range of fields as it fits into different aspects of the design–build–test cycle typical of modern research. **MetQy** could inform experimental design in synthetic and systems biology studies, demonstrated in Chapter 5, as well as facilitate the analysis of data generated by incorporating biochemical and genomic information to experimental data. By combining **MetQy** with statistical tools and models, such as vector similarity or machine learning (Keith, 2017), new predictions and insights could be gained from the in-built KEGG data, such as identifying organisms previously unknown to be electroactive. Finally, these two tools, the electrochemical platform and **MetQy**, could be used in the same design–build–test cycle to identify potential organisms with electroactive capabilities and to characterise them, feeding back into the cycle using the information gained to further our understanding of EET mechanisms.

# References

- Alberts B, Johnson A, Lewis J, Raff M, Roberts K and Walter P. 2002. Molecular biology of the cell. Garland Science. ISBN 0815332181
- Alberty RA. 2001. Half Reactions as a Function of pH and Ionic Strength. *Archives of biochemistry and biophysics*, 389(1):94–109. ISSN 00039861. doi:10.1006/abbi.2001.2318
- Alberty RA. 2003. Thermodynamics of Biochemical Reactions. Hoboken, NJ, USA: John Wiley & Sons, Inc. ISBN 9780471332602. doi:10.1002/0471332607
- Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W and Lipman DJ. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–402. ISSN 0305-1048
- Anthony C. 1988. Bacterial energy transduction. Academic Press. ISBN 0120588153
- Awate B, Steidl RJ, Hamlisher T and Reguera G. 2017. Stimulation of electro-fermentation in single-chamber microbial electrolysis cells driven by genetically engineered anode biofilms. *Journal of Power Sources*, 356(356):510–518. ISSN 03787753. doi:10.1016/j.jpowsour.2017.02.053
- Babauta J, Renslow R, Lewandowski Z and Beyenal H. 2012. Electrochemically active biofilms: Facts and fiction. A review. *Biofouling*, 28(8):789–812. ISSN 08927014. doi:10.1080/08927014.2012.710324
- Balagaddé FK, Song H, Ozaki J, Collins CH, Barnet M, Arnold FH, Quake SR and You L. 2008. A synthetic *Escherichia coli* predator-prey ecosystem. *Molecular Systems Biology*, 4(187):187. ISSN 17444292. doi:10.1038/msb.2008.24
- Bar-Even A, Flamholz A, Noor E and Milo R. 2012. Thermodynamic constraints shape the structure of carbon fixation pathways. *Biochimica et Biophysica Acta - Bioenergetics*, 1817(9):1646–1659. ISSN 00052728. doi:10.1016/j.bbabi.2012.05.002

- Barua S and Dhar BR. 2017. Advances towards understanding and engineering direct inter-species electron transfer in anaerobic digestion. *Bioresource Technology*, 244(July):698–707. ISSN 18732976. doi:10.1016/j.biortech.2017.08.023
- Bateman A, Martin MJ, O'Donovan C, Magrane M, Alpi E, Antunes R, Bely B, Bingley M, Bonilla C, Britto R, Bursteinas B, Bye-A-Jee H, Cowley A, Silva AD, Giorgi MD, Dogan T, Fazzini F, Castro LG, Figueira L, Garmiri P, Georghiou G, Gonzalez D, Hatton-Ellis E, Li W, Liu W, Lopez R, Luo J, Lussi Y, MacDougall A, Nightingale A, Palka B, Pichler K, Poggioli D, Pundir S, Pureza L, Qi G, Renaux A, Rosanoff S, Saidi R, Sawford T, Shypitsyna A, Speretta E, Turner E, Tyagi N, Volynkin V, Wardell T, Warner K, Watkins X, Zaru R, Zellner H, Xenarios I, Bougueleret L, Bridge A, Poux S, Redaschi N, Aimo L, Argoud-Puy G, Auchincloss A, Axelsen K, Bansal P, Baratin D, Blatter MC, Boeckmann B, Bolleman J, Boutet E, Breuza L, Casal-Casas C, de Castro E, Coudert E, Cuche B, Doche M, Dornevil D, Duvaud S, Estreicher A, Famiglietti L, Feuermann M, Gasteiger E, Gehant S, Gerritsen V, Gos A, Gruaz-Gumowski N, Hinz U, Hulo C, Jungo F, Keller G, Lara V, Lemercier P, Lieberherr D, Lombardot T, Martin X, Masson P, Morgat A, Neto T, Noupikel N, Paesano S, Pedruzzi I, Pilbout S, Pozzato M, Pruess M, Rivoire C, Roechert B, Schneider M, Sigrist C, Sonesson K, Staehli S, Stutz A, Sundaram S, Tognolli M, Verbregue L, Veuthey AL, Wu CH, Arighi CN, Arminski L, Chen C, Chen Y, Garavelli JS, Huang H, Laiho K, McGarvey P, Natale DA, Ross K, Vinayaka CR, Wang Q, Wang Y, Yeh LS and Zhang J. 2017. UniProt: the universal protein knowledgebase. *Nucleic Acids Research*, 45(D1):D158–D169. ISSN 0305-1048. doi:10.1093/nar/gkw1099
- Bauer M and Kulinsky L. 2018. Fabrication of a lab-on-chip device using material extrusion (3D printing) and demonstration via malaria-Ab ELISA. *Micromachines*, 9(1). ISSN 2072666X. doi:10.3390/mi9010027
- Beard DA. 2004. Thermodynamic-based computational profiling of cellular regulatory control in hepatocyte metabolism. *AJP: Endocrinology and Metabolism*, 288(3):E633–E644. ISSN 0193-1849. doi:10.1152/ajpendo.00239.2004
- Beckmann S, Welte C, Li X, Oo YM, Kroeninger L, Heo Y, Zhang M, Ribeiro D, Lee M, Bhadbhade M, Marjo CE, Seidel J, Deppenmeier U and Manefield M. 2016. Novel phenazine crystals enable direct electron transfer to methanogens in anaerobic digestion by redox potential modulation. *Energy and Environmental Science*, 9(2):644–655. ISSN 17545706. doi:10.1039/c5ee03085d
- Bennetto H. 1984. Microbial fuel cells. In Michelson A and Bannister J, (eds.) *Life Chemistry Reports*, volume 2, 363–453. London, UK: Harwood Academic

- Benson DA, Cavanaugh M, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J and Sayers EW. 2012. GenBank. *Nucleic Acids Research*, 41(D1):D36–D42. doi:10.1093/nar/gks1195
- Biffinger JC, Fitzgerald LA, Ray R, Little BJ, Lizewski SE, Petersen ER, Ringeisen BR, Sanders WC, Sheehan PE, Pietron JJ, Baldwin JW, Nadeau LJ, Johnson GR, Ribbens M, Finkel SE and Nealson KH. 2011. The utility of *Shewanella japonica* for microbial fuel cells. *Bioresource Technology*, 102(1):290–297. ISSN 09608524. doi:10.1016/j.biortech.2010.06.078
- Bond DR and Lovley DR. 2003. Electricity Production by *Geobacter sulfurreducens* Attached to Electrodes Electricity Production by *Geobacter sulfurreducens* Attached to Electrodes. *Applied and Environmental Microbiology*, 69(3):1548–1555. ISSN 00992240. doi:10.1128/AEM.69.3.1548
- Bono H, Ogata H, Goto S and Kanehisa M. 1998. Reconstruction of amino acid biosynthesis pathways from the complete genome sequence. *Genome Research*, 8(3):203–210. ISSN 10889051. doi:10.1101/gr.8.3.203
- Bose A, Gardel E, Vidoudez C, Parra E and Girguis P. 2014. Electron uptake by iron-oxidizing phototrophic bacteria. *Nature Communications*, 5(1):3391. ISSN 2041-1723. doi:10.1038/ncomms4391
- Bryant MP, Wolin EA, Wolin MJ and Wolfe RS. 1967. Methanobacillus omelianskii, a symbiotic association of two species of bacteria. *Archiv fur Mikrobiologie*, 59(1):20–31. ISSN 0003-9276
- Buonomenna M. 2016. Microbial Electrosynthesis of Methane: Challenges and Recent Progresses. *Current Biochemical Engineering*, 3(3):235–250. ISSN 22127119. doi:10.2174/1570180813666160527115440
- Butler JE, Young ND and Lovley DR. 2010. Evolution of electron transfer out of the cell: Comparative genomics of six *Geobacter* genomes. *BMC Genomics*, 11(1):40. ISSN 14712164. doi:10.1186/1471-2164-11-40
- Carson MA and Basiliko N. 2016. Approaches to R education in Canadian universities. *F1000Research*, 5:2802. ISSN 2046-1402. doi:10.12688/f1000research.10232.1
- Caspi R, Altman T, Dreher K, Fulcher CA, Subhraveti P, Keseler IM, Kothari A, Krummenacker M, Latendresse M, Mueller LA, Ong Q, Paley S, Pujar A, Shearer AG, Travers M, Weerasinghe D, Zhang P and Karp PD. 2012. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic acids research*, 40:D742–53. ISSN 1362-4962. doi:10.1093/nar/gkr1014

- Chae MP, Rozen WM, McMenamin PG, Findlay MW, Spychal RT and Hunter-Smith DJ. 2015. Emerging Applications of Bedside 3D Printing in Plastic Surgery. *Frontiers in surgery*, 2:25. ISSN 2296-875X. doi:10.3389/fsurg.2015.00025
- Chang IS, Moon HS, Bretschger O, Jang JK, Park HI, Neilson KH and Kim BH. 2006. Electrochemical active bacteria (EAB) and mediator-less microbial fuel cells. *Journal of Microbiology and Biotechnology*, 16(3)
- Chen J, Zhang L, Hu Y, Huang W, Niu Z and Sun J. 2017. Bacterial community shift and incurred performance in response to in situ microbial self-assembly graphene and polarity reversion in microbial fuel cell. *Bioresource Technology*, 241:220–227. ISSN 0960-8524. doi:10.1016/J.BIORTECH.2017.05.123
- Cheng S and Logan BE. 2007. Sustainable and efficient biohydrogen production via electrohydrogenesis. *Proceedings of the National Academy of Sciences*, 104(47):18871–18873. ISSN 0027-8424. doi:10.1073/pnas.0706379104
- Chong GW, Karbelkar AA and El-Naggar MY. 2018. Nature’s conductors: what can microbial multi-heme cytochromes teach us about electron transport and biological energy conversion? *Current Opinion in Chemical Biology*, 47:7–17. ISSN 1367-5931. doi:10.1016/J.CBPA.2018.06.007
- Cohen G. 2014. Microbial Biochemistry. Dordrecht: Springer. ISBN 9789401789073
- Costa KC, Lie TJ, Jacobs MA and Leigh JA. 2013. H<sub>2</sub>-independent growth of the hydrogenotrophic methanogen *Methanococcus maripaludis*. *mBio*, 4(2):e00062–13. ISSN 2150-7511. doi:10.1128/mBio.00062-13
- Cotterill S, Heidrich E and Curtis T. 2015. Microbial Electrolysis Cells for Hydrogen Production, volume 9. Elsevier Ltd. ISBN 9781782423966. doi:10.1016/B978-1-78242-375-1.00009-5
- Croese E, Pereira MA, Euverink GJW, Stams AJM and Geelhoed JS. 2011. Analysis of the microbial community of the biocathode of a hydrogen-producing microbial electrolysis cell. *Applied Microbiology and Biotechnology*, 92(5):1083–1093. ISSN 01757598. doi:10.1007/s00253-011-3583-x
- Das S, Chatterjee P and Ghangrekar MM. 2018. Increasing methane content in biogas and simultaneous value added product recovery using microbial electrosynthesis. *Water Science and Technology*, 77(5):1293–1302. ISSN 02731223. doi:10.2166/wst.2018.002
- De Martino D, Capuani F and De Martino A. 2014. Inferring metabolic phenotypes from the exometabolome through a thermodynamic variational principle. *New Journal of Physics*, 16(11):115018. ISSN 13672630. doi:10.1088/1367-2630/16/11/115018

- Deutzmann JS, Sahin M and Spormann AM. 2015. Extracellular Enzymes Facilitate Electron Uptake in Biocorrosion and Bioelectrosynthesis. *mBio*, 6(2):e00496–15. ISSN 2150-7511. doi:10.1128/MBIO.00496-15
- Dewan A, Beyenal H and Lewandowski Z. 2008. Scaling up Microbial Fuel Cells. *Environmental Science & Technology*, 42(20):7643–7648. ISSN 0013-936X. doi:10.1021/es800775d
- Ditzig J, Liu H and Logan BE. 2007. Production of hydrogen from domestic wastewater using a bioelectrochemically assisted microbial reactor (BEAMR). *International Journal of Hydrogen Energy*, 32(13):2296–2304. ISSN 03603199. doi:10.1016/j.ijhydene.2007.02.035
- Doelle HW. 1975. Bacterial metabolism. Academic Press. ISBN 0122193520
- Doelle HW, Rokem S and Berovic M, (eds.) . 2009. Biotechnology Volume V. UNESCO-EOLSS. ISBN 1848262590
- D’Souza G, Waschina S, Pande S, Bohl K, Kaleta C and Kost C. 2014. Less is more: Selective advantages can explain the prevalent loss of biosynthetic genes in bacteria. *Evolution*, 68(9):2559–2570. ISSN 15585646. doi:10.1111/evo.12468
- Falkowski PG, Fenchel T and Delong EF. 2008. The microbial engines that drive earth’s biogeochemical cycles. *Science*, 320(5879):1034–1039. ISSN 00368075. doi:10.1126/science.1153213
- Fan Y, Xu S, Schaller R, Jiao J, Chaplen F and Liu H. 2011. Nanoparticle decorated anodes for enhanced current generation in microbial electrochemical cells. *Biosensors and Bioelectronics*, 26(5):1908–1912. ISSN 09565663. doi:10.1016/j.bios.2010.05.006
- Farrell F, Soyer OS and Quince C. 2018. Machine learning based prediction of functional capabilities in metagenomically assembled microbial genomes. *bioRxiv*, 307157. doi:10.1101/307157
- Freilich S, Zarecki R, Eilam O, Segal ES, Henry CS, Kupiec M, Gophna U, Sharan R and Ruppin E. 2011. Competitive and cooperative metabolic interactions in bacterial communities. *Nature Communications*, 2(1):589. ISSN 20411723. doi:10.1038/ncomms1597
- Friman H, Schechter A, Nitzan Y and Cahan R. 2012. Effect of external voltage on *Pseudomonas putida* F1 in a bio electrochemical cell using toluene as sole carbon and energy source. *Microbiology*, 158(2):414–423. ISSN 1350-0872. doi:10.1099/mic.0.053298-0
- Gajda I, Greenman J, Melhuish C and Ieropoulos IA. 2015. Simultaneous electricity generation and microbially-assisted electrosynthesis in ceramic MFCs. *Bioelectrochemistry*, 104:58–64. ISSN 1878562X. doi:10.1016/j.bioelechem.2015.03.001

- Gajda I, Greenman J, Melhuish C, Santoro C, Li B, Cristiani P and Ieropoulos IA. 2014. Water formation at the cathode and sodium recovery using Microbial Fuel Cells (MFCs). *Sustainable Energy Technologies and Assessments*, 7:187–194. ISSN 22131388. doi:10.1016/j.seta.2014.05.001
- Gajda I, Greenman J, Santoro C, Serov A, Atanasov P and Ieropoulos IA. 2018. Small Ceramic Microbial Fuel Cell As a Trigenerative System for Electricity, Organics Degradation and Urine Filtration. *Meeting Abstracts*, MA2018-02(27):907–907. ISSN 2151-2041
- Galperin MY, Makarova KS, Wolf YI and Koonin EV. 2015. Expanded microbial genome coverage and improved protein family annotation in the COG database. *Nucleic acids research*, 43:D261–9. ISSN 1362-4962. doi:10.1093/nar/gku1223
- Gamry Instruments. 2012. Quick Check of EIS System Performance. *Gamry Application Note*, 1–11
- Ghangrekar MM and Chatterjee P. 2017. A Systematic Review on Bioelectrochemical Systems Research. *Current Pollution Reports*, 3(4):281–288. ISSN 2198-6592. doi:10.1007/s40726-017-0071-7
- Gimkiewicz C and Harnisch F. 2013. Waste Water Derived Electroactive Microbial Biofilms: Growth, Maintenance, and Basic Characterization. *Journal of Visualized Experiments*, (82):50800. ISSN 1940-087X. doi:10.3791/50800
- González Del Campo A, Cañizares P, Lobato J, Rodrigo M and Fernandez Morales F. 2016. Effects of External Resistance on Microbial Fuel Cell's Performance. In *Handbook of Environmental Chemistry*, volume 34, 175–197. Springer, Cham. ISBN 978-3-319-17100-5. doi:10.1007/698-2014-290
- Goodman B and Gardner H. 2018. The microbiome and cancer. *Journal of Pathology*, 244(5):667–676. ISSN 10969896. doi:10.1002/path.5047
- Goyal N, Zhou Z and Karimi IA. 2016. Metabolic processes of *Methanococcus maripaludis* and potential applications. *Microbial Cell Factories*, 15(1):107. ISSN 1475-2859. doi:10.1186/s12934-016-0500-0
- Green JL, Bohannan BJ and Whitaker RJ. 2008. Microbial Biogeography : From Taxonomy to Traits. *Science*, 320:1039–1044. doi:10.1126/science.1153475
- Großkopf T and Soyer OS. 2014. Synthetic microbial communities. *Current Opinion in Microbiology*, 18(1):72–77. ISSN 18790364. doi:10.1016/j.mib.2014.02.002

- Großkopf T and Soyer OS. 2016. Microbial diversity arising from thermodynamic constraints. *ISME Journal*, 10(11):2725–2733. ISSN 17517370. doi:10.1038/ismej.2016.49
- Großkopf T, Zenobi S, Alston M, Folkes L, Swarbreck D and Soyer OS. 2016. A stable genetic polymorphism underpinning microbial syntrophy. *ISME Journal*, 10(12):2844–2853. ISSN 17517370. doi:10.1038/ismej.2016.80
- Gude VG. 2015. Microbial Fuel Cells for Wastewater Treatment and Energy Generation. Elsevier Ltd. ISBN 9781782423966. doi:10.1016/B978-1-78242-375-1.00008-3
- Habermann W and Pommer E. 1991. Biological fuel cells with sulphide storage capacity. *Applied Microbiology and Biotechnology*, 35(1):128–133. ISSN 0175-7598. doi:10.1007/BF00180650
- Hamilton JJ, Calixto Contreras M and Reed JL. 2015. Thermodynamics and H<sub>2</sub> Transfer in a Methanogenic, Syntrophic Community. *PLoS Computational Biology*, 11(7):e1004364. ISSN 15537358. doi:10.1371/journal.pcbi.1004364
- Han A, Hou H, Li L, Kim HS and de Figueiredo P. 2013. Microfabricated devices in microbial bioenergy sciences. *Trends in Biotechnology*, 31(4):225–232. ISSN 01677799. doi:10.1016/j.tibtech.2012.12.002
- Hartel P. 2005. MICROBIAL PROCESSES — Environmental Factors. *Encyclopedia of Soils in the Environment*, 448–455. doi:10.1016/B0-12-348530-4/00155-7
- Heidelberg JF, Seshadri R, Haveman SA, Hemme CL, Paulsen IT, Kolonay JF, Eisen JA, Ward N, Methe B, Brinkac LM, Daugherty SC, Deboy RT, Dodson RJ, Durkin AS, Madupu R, Nelson WC, Sullivan SA, Fouts D, Haft DH, Selengut J, Peterson JD, Davidsen TM, Zafar N, Zhou L, Radune D, Dimitrov G, Hance M, Tran K, Khouri H, Gill J, Utterback TR, Feldblyum TV, Wall JD, Voordouw G and Fraser CM. 2004. The genome sequence of the anaerobic, sulfate-reducing bacterium *Desulfovibrio vulgaris* Hildenborough. *Nature Biotechnology*, 22(5):554–559. ISSN 1087-0156. doi:10.1038/nbt959
- Hendrickson EL, Kaul R, Zhou Y, Bovee D, Chapman P, Chung J, Conway de Macario E, Dodsworth JA, Gillett W, Graham DE, Hackett M, Haydock AK, Kang A, Land ML, Levy R, Lie TJ, Major TA, Moore BC, Porat I, Palmeiri A, Rouse G, Saenphimmachak C, Soll D, Van Dien S, Wang T, Whitman WB, Xia Q, Zhang Y, Larimer FW, Olson MV and Leigh JA. 2004. Complete Genome Sequence of the Genetically Tractable Hydrogenotrophic Methanogen *Methanococcus maripaludis*. *Journal of Bacteriology*, 186(20):6956–6969. ISSN 0021-9193. doi:10.1128/JB.186.20.6956-6969.2004



- Hendrickx L, De Wever H, Hermans V, Mastroleo F, Morin N, Wilmotte A, Janssen P and Mergeay M. 2006. Microbial ecology of the closed artificial ecosystem MELiSSA (Micro-Ecological Life Support System Alternative): Reinventing and compartmentalizing the Earth's food and oxygen regeneration system for long-haul space exploration missions. *Research in Microbiology*, 157(1):77–86. ISSN 09232508. doi:10.1016/j.resmic.2005.06.014
- Hendrickx L and Mergeay M. 2007. From the deep sea to the stars: human life support through minimal communities. *Current Opinion in Microbiology*, 10(3):231–237. ISSN 13695274. doi:10.1016/j.mib.2007.05.007
- Henry CS, Broadbelt LJ and Hatzimanikatis V. 2007. Thermodynamics-based metabolic flux analysis. *Biophysical Journal*, 92(5):1792–1805. ISSN 00063495. doi:10.1529/biophysj.106.093138
- Henry CS, Jankowski MD, Broadbelt LJ and Hatzimanikatis V. 2006. Genome-scale thermodynamic analysis of *Escherichia coli* metabolism. *Biophysical Journal*, 90(4):1453–1461. ISSN 00063495. doi:10.1529/biophysj.105.071720
- Hill EA, Chrisler WB, Beliaev AS and Bernstein HC. 2017. A flexible microbial co-culture platform for simultaneous utilization of methane and carbon dioxide from gas feedstocks. *Bioresource Technology*, 228:250–256. ISSN 0960-8524. doi:10.1016/J.BIORTECH.2016.12.111
- Hodgson DM, Smith A, Dahale S, Stratford JP, Li JV, Grüning A, Bushell ME, Marchesi JR and Avignone-Rossa C. 2016. Segregation of the Anodic Microbial Communities in a Microbial Fuel Cell Cascade. *Frontiers in Microbiology*, 7:699. ISSN 1664-302X. doi:10.3389/fmicb.2016.00699
- Hong G and Pachter R. 2016. Bound Flavin-Cytochrome Model of Extracellular Electron Transfer in *Shewanella oneidensis*: Analysis by Free Energy Molecular Dynamics Simulations. *Journal of Physical Chemistry B*, 120(25):5617–5624. ISSN 15205207. doi:10.1021/acs.jpcc.6b03851
- Hou C, Yang D, Liang B and Liu A. 2014. Enhanced performance of a glucose/O<sub>2</sub> biofuel cell assembled with laccase-covalently immobilized three-dimensional macroporous gold film-based biocathode and bacterial surface displayed glucose dehydrogenase-based bioanode. *Analytical Chemistry*, 86(12):6057–6063. ISSN 15206882. doi:10.1021/ac501203n
- Hou J, Scalcinati G, Oldiges M and Vemuri GN. 2010. Metabolic impact of increased NADH availability in *Saccharomyces cerevisiae*. *Applied and Environmental Microbiology*, 76(3):851–859. ISSN 00992240. doi:10.1128/AEM.02040-09
- Hu Z. 2008. Electricity generation by a baffle-chamber membraneless microbial fuel cell. *Journal of Power Sources*, 179(1):27–33. ISSN 03787753. doi:10.1016/j.jpowsour.2007.12.094

- Huang L, Jiang L, Wang Q, Quan X, Yang J and Chen L. 2014. Cobalt recovery with simultaneous methane and acetate production in biocathode microbial electrolysis cells. *Chemical Engineering Journal*, 253:281–290. ISSN 13858947. doi:10.1016/j.cej.2014.05.080
- Ieropoulos I, Winfield J, Gajda I, Walter A, Papaharalabos G, Jimenez IM, Pasternak G, You J, Tremouli A, Stinchcombe A, Forbes S and Greenman J. 2015. The Practical Implementation of Microbial Fuel Cell Technology. Elsevier Ltd. ISBN 9781782423966. doi:10.1016/B978-1-78242-375-1.00012-5
- Ieropoulos IA and Greenman J. 2018. Microbial fuel cell, method of controlling and measuring the redox potential difference of the fuel cell
- Ieropoulos IA, Stinchcombe A, Gajda I, Forbes S, Merino-Jimenez I, Pasternak G, Sanchez-Herranz D and Greenman J. 2016. Pee power urinal – microbial fuel cell technology field trials in the context of sanitation. *Environmental Science: Water Research and Technology*, 2(2):336–343. ISSN 20531419. doi:10.1039/c5ew00270b
- Jablonski S, Rodowicz P and Lukaszewicz M. 2015. Methanogenic archaea database containing physiological and biochemical characteristics. *International Journal of Systematic and Evolutionary Microbiology*, 65(4):1360–1368. doi:10.1099/ij.s.0.000065
- Jain A, Zhang X, Pastorella G, Connolly JO, Barry N, Woolley R, Krishnamurthy S and Marsili E. 2012. Electron transfer mechanism in *Shewanella loihica* PV-4 biofilms formed at graphite electrode. *Bioelectrochemistry*, 87:28–32. ISSN 15675394. doi:10.1016/j.bioelechem.2011.12.012
- Jiang Y, Su M and Li D. 2014. Removal of sulfide and production of methane from carbon dioxide in microbial fuel cells-microbial electrolysis cell (MFCs-MEC) coupled system. *Applied Biochemistry and Biotechnology*, 172(5):2720–2731. ISSN 15590291. doi:10.1007/s12010-013-0718-9
- Jolliffe IT. 2010. Principal component analysis. Springer. ISBN 9781441929990
- Kadier A, Kalil MS, Abdesahian P, Chandrasekhar K, Mohamed A, Azman NF, Logroño W, Simayi Y and Hamid AA. 2016a. Recent advances and emerging challenges in microbial electrolysis cells (MECs) for microbial production of hydrogen and value-added chemicals. *Renewable and Sustainable Energy Reviews*, 61:501–525. ISSN 18790690. doi:10.1016/j.rser.2016.04.017
- Kadier A, Simayi Y, Abdesahian P, Azman NF, Chandrasekhar K and Kalil MS. 2016b. A comprehensive review of microbial electrolysis cells (MEC) reactor designs and configurations for sustainable hydrogen gas production. *Alexandria Engineering Journal*, 55(1):427–443. ISSN 1110-0168. doi:10.1016/J.AEJ.2015.10.008

- Kane AL, Bond DR and Gralnick JA. 2013. Electrochemical Analysis of *Shewanella oneidensis* Engineered To Bind Gold Electrodes. *ACS Synthetic Biology*, 2(2):93–101. ISSN 2161-5063. doi:10.1021/sb300042w
- Kanehisa M. 1997. A database for post-genome analysis. *Trends in Genetics*, 13(9):375–376. ISSN 01689525. doi:10.1016/S0168-9525(97)01223-7
- Kanehisa M. 2004. The KEGG resource for deciphering the genome. *Nucleic Acids Research*, 32(90001):277D–280. ISSN 1362-4962. doi:10.1093/nar/gkh063
- Kanehisa M. 2013. Chemical and genomic evolution of enzyme-catalyzed reaction networks. *FEBS Letters*, 587(17):2731–2737. ISSN 00145793. doi:10.1016/j.febslet.2013.06.026
- Kanehisa M, Furumichi M, Tanabe M, Sato Y and Morishima K. 2017. KEGG: New perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research*, 45(D1):D353–D361. ISSN 13624962. doi:10.1093/nar/gkw1092
- Kanehisa M and Goto S. 2000. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 27(1):29–34. ISSN 03051048. doi:10.1093/nar/27.1.29
- Kanehisa M, Goto S, Sato Y, Furumichi M and Tanabe M. 2012. KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, 40(D1):D109–D114. ISSN 03051048. doi:10.1093/nar/gkr988
- Kanehisa M, Goto S, Sato Y, Kawashima M, Furumichi M and Tanabe M. 2014. Data, information, knowledge and principle: Back to metabolism in KEGG. *Nucleic Acids Research*, 42(D1):D199–205. ISSN 03051048. doi:10.1093/nar/gkt1076
- Kanehisa M, Sato Y, Kawashima M, Furumichi M and Tanabe M. 2016a. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Research*, 44(D1):D457–D462. ISSN 13624962. doi:10.1093/nar/gkv1070
- Kanehisa M, Sato Y and Morishima K. 2016b. BlastKOALA and GhostKOALA: KEGG Tools for Functional Characterization of Genome and Metagenome Sequences. *Journal of Molecular Biology*, 428(4):726–731. ISSN 10898638. doi:10.1016/j.jmb.2015.11.006
- Kato S, Hashimoto K and Watanabe K. 2012. Microbial interspecies electron transfer via electric currents through conductive minerals. *Proceedings of the National Academy of Sciences*, 109(25):10042–10046. ISSN 0027-8424. doi:10.1073/pnas.1117592109
- Keith JM, (ed.) . 2017. Bioinformatics: Volume I: Data, Sequence Analysis, and Evolution, volume 1525 of *Methods in Molecular Biology*. New York, NY: Springer. ISBN 978-1-4939-6620-2. doi:10.1007/978-1-4939-6622-6

- Kerner A, Park J, Williams A and Lin XN. 2012. A programmable escherichia coli consortium via tunable symbiosis. *PLoS ONE*, 7(3):e34032. ISSN 19326203. doi:10.1371/journal.pone.0034032
- Kim HJ, Park HS, Hyun MS, Chang IS, Kim M and Kim BH. 2002. A mediator-less microbial fuel cell using a metal reducing bacterium, *Shewanella putrefaciens*. *Enzyme and Microbial Technology*, 30(2):145–152. ISSN 01410229. doi:10.1016/S0141-0229(01)00478-1
- Kim JR, Cheng S, Oh SE and Logan BE. 2007. Power generation using different cation, anion, and ultrafiltration membranes in microbial fuel cells. *Environmental Science and Technology*, 41(3):1004–9. ISSN 0013-936X. doi:10.1021/es062202m
- Kouzuma A, Kato S and Watanabe K. 2015a. Microbial interspecies interactions: recent findings in syntrophic consortia. *Frontiers in microbiology*, 6:477. ISSN 1664-302X. doi:10.3389/fmicb.2015.00477
- Kouzuma A, Kato S and Watanabe K. 2015b. Microbial interspecies interactions: Recent findings in syntrophic consortia. *Frontiers in Microbiology*, 6(MAY):477. ISSN 1664302X. doi:10.3389/fmicb.2015.00477
- Kouzuma A, Meng XY, Kimura N, Hashimoto K and Watanabe K. 2010. Disruption of the putative cell surface polysaccharide biosynthesis gene SO3177 in *Shewanella oneidensis* MR-1 enhances adhesion to electrodes and current generation in microbial fuel cells. *Applied and environmental microbiology*, 76(13):4151–7. ISSN 1098-5336. doi:10.1128/AEM.00117-10
- Kracke F. 2016. Understanding extracellular electron transport of industrial microorganisms and optimization for production application. Ph.D. thesis, The University of Queensland. doi:10.14264/uql.2016.202
- Kumar G, Saratale RG, Kadier A, Sivagurunathan P, Zhen G, Kim SH and Saratale GD. 2017. A review on bio-electrochemical systems (BESs) for the syngas and value added biochemicals production. *Chemosphere*, 177:84–92. ISSN 18791298. doi:10.1016/j.chemosphere.2017.02.135
- Kumar R, Singh L, Wahid ZA and Din MFM. 2015. Exoelectrogens in microbial fuel cells toward bioelectricity generation: A review. *International Journal of Energy Research*, 39(8):1048–1067. ISSN 1099114X. doi:10.1002/er.3305
- Langille MG, Zaneveld J, Caporaso JG, McDonald D, Knights D, Reyes JA, Clemente JC, Burkepille DE, Vega Thurber RL, Knight R, Beiko RG and Huttenhower C. 2013. Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. *Nature Biotechnology*, 31(9):814–821. ISSN 10870156. doi:10.1038/nbt.2676

- Lee WJ and Hase K. 2014. Gut microbiota-generated metabolites in animal health and disease. *Nature Chemical Biology*, 10(6):416–424. ISSN 15524469. doi:10.1038/nchembio.1535
- Lefrou C, Fabry P and Poignet JC. 2012. Electrochemistry. The Basics, With Examples. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-30249-7. doi:10.1007/978-3-642-30250-3
- Lie TJ, Costa KC, Lupa B, Korpole S, Whitman WB and Leigh JA. 2012. Essential anaplerotic role for the energy-converting hydrogenase Eha in hydrogenotrophic methanogenesis. *Proceedings of the National Academy of Sciences of the United States of America*, 109(38):15473–8. ISSN 1091-6490. doi:10.1073/pnas.1208779109
- Liu H, Cheng S and Logan BE. 2005. Production of electricity from acetate or butyrate using a single-chamber microbial fuel cell. *Environmental Science and Technology*, 39(2):658–662. ISSN 0013936X. doi:10.1021/es048927c
- Liu H and Logan BE. 2004. Electricity generation using an air-cathode single chamber microbial fuel cell in the presence and absence of a proton exchange membrane. *Environmental Science and Technology*, 38(14):4040–4046. ISSN 0013936X. doi:10.1021/es0499344
- Liu H, Ramnarayanan R and Logan BE. 2004. Production of Electricity during Wastewater Treatment Using a Single Chamber Microbial Fuel Cell. *Environmental Science and Technology*, 38(7):2281–2285. ISSN 0013936X. doi:10.1021/es034923g
- Liu Y, Wang Z, Liu J, Levar C, Edwards MJ, Babauta JT, Kennedy DW, Shi Z, Beyenal H, Bond DR, Clarke TA, Butt JN, Richardson DJ, Rosso KM, Zachara JM, Fredrickson JK and Shi L. 2014. A trans-outer membrane porin-cytochrome protein complex for extracellular electron transfer by *Geobacter sulfurreducens* PCA. *Environmental Microbiology Reports*, 6(6):776–785. doi:10.1111/1758-2229.12204
- Lof M, Janus MM and Krom BP. 2017. Metabolic Interactions between Bacteria and Fungi in Commensal Oral Biofilms. *Journal of fungi (Basel, Switzerland)*, 3(3). ISSN 2309-608X. doi:10.3390/jof3030040
- Logan BE. 2009. Exoelectrogenic bacteria that power microbial fuel cells. *Nature Reviews Microbiology*, 7(5):375–381. ISSN 1740-1526. doi:10.1038/nrmicro2113
- Logan BE, Hamelers B, Rozendal R, Schröder U, Keller J, Freguia S, Aelterman P, Verstraete W and Rabaey K. 2006. Microbial fuel cells: Methodology and technology. *Environmental Science and Technology*, 40(17):5181–5192. ISSN 0013936X. doi:10.1021/es0605016

- Logan BE and Rabaey K. 2012. Conversion of wastes into bioelectricity and chemicals by using microbial electrochemical technologies. *Science (New York, N.Y.)*, 337(6095):686–90. ISSN 1095-9203. doi:10.1126/science.1217412
- Logan BE and Regan JM. 2006. Electricity-producing bacterial communities in microbial fuel cells. *Trends in Microbiology*, 14(12):512–518. ISSN 0966842X. doi:10.1016/j.tim.2006.10.003
- Lohner ST, Deutzmann JS, Logan BE, Leigh J and Spormann AM. 2014. Hydrogenase-independent uptake and metabolism of electrons by the archaeon *Methanococcus maripaludis*. *ISME Journal*, 8(8):1673–1681. ISSN 17517370. doi:10.1038/ismej.2014.82
- Louca S, Jacques SM, Pires AP, Leal JS, González AL, Doebeli M and Farjalla VF. 2017. Functional structure of the bromeliad tank microbiome is strongly shaped by local geochemical conditions. *Environmental Microbiology*, 19(8):3132–3151. ISSN 14622920. doi:10.1111/1462-2920.13788
- Louca S, Parfrey LW and Doebeli M. 2016. Decoupling function and taxonomy in the global ocean microbiome. *Science*, 353(6305):1272–7. doi:10.1126/science.aaf4507
- Lovley DR. 2017a. Happy together: Microbial communities that hook up to swap electrons. *ISME Journal*, 11(2):327–336. ISSN 17517370. doi:10.1038/ismej.2016.136
- Lovley DR. 2017b. Syntrophy Goes Electric: Direct Interspecies Electron Transfer. *Annual Review of Microbiology*, 71(1):annurev-micro-030117-020420. ISSN 0066-4227. doi:10.1146/annurev-micro-030117-020420
- Lu L and Ren ZJ. 2016. Microbial electrolysis cells for waste biorefinery: A state of the art review. *Bioresource Technology*, 215:254–264. ISSN 18732976. doi:10.1016/j.biortech.2016.03.034
- Luo W and Brouwer C. 2013. Pathview: an R/Bioconductor package for pathway-based data integration and visualization. *Bioinformatics*, 29(14):1830–1831. ISSN 1460-2059. doi:10.1093/bioinformatics/btt285
- Manohar AK, Bretschger O, Nealson KH and Mansfeld F. 2008. The polarization behavior of the anode in a microbial fuel cell. *Electrochimica Acta*, 53(9):3508–3513. ISSN 00134686. doi:10.1016/j.electacta.2007.12.002
- Marsili E, Rollefson JB, Baron DB, Hozalski RM and Bond DR. 2008. Microbial biofilm voltammetry: Direct electrochemical characterization of catalytic electrode-attached biofilms. *Applied and Environmental Microbiology*, 74(23):7329–7337. ISSN 00992240. doi:10.1128/AEM.00177-08

- Martinez-Vernon AS, Farrell F and Soyer OS. 2018. MetQy - an R package to query metabolic functions of genes and genomes. *Bioinformatics*, 34(23). ISSN 1367-4803. doi:10.1093/bioinformatics/bty447
- Martiny JB, Jones SE, Lennon JT and Martiny AC. 2015. Microbiomes in light of traits: A phylogenetic perspective. *Science*, 350(6261). ISSN 10959203. doi:10.1126/science.aac9323
- McAnulty MJ, Poosarla VG, Kim KY, Jasso-Chávez R, Logan BE and Wood TK. 2017. Electricity from methane by reversing methanogenesis. *Nature Communications*, 8(May). ISSN 20411723. doi:10.1038/ncomms15419
- McDonald AG, Boyce S and Tipton KF. 2009. ExplorEnz: The primary source of the IUBMB enzyme list. *Nucleic Acids Research*, 37(SUPPL. 1):D593–7. ISSN 03051048. doi:10.1093/nar/gkn582
- Merino Jimenez I, Greenman J and Ieropoulos IA. 2017. Electricity and catholyte production from ceramic MFCs treating urine. *International Journal of Hydrogen Energy*, 42(3):1791–1799. ISSN 03603199. doi:10.1016/j.ijhydene.2016.09.163
- Meyer HP, Minas W and Schmidhalter D. 2016. Industrial-Scale Fermentation. In *Industrial Biotechnology*, 1–53. Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA. doi:10.1002/9783527807833.ch1
- Miller RB, Sadek A, Rodriguez A, Iannuzzi M, Gai C, Senko JM and Monty CN. 2016. Use of an electrochemical split cell technique to evaluate the influence of *Shewanella oneidensis* activities on corrosion of carbon steel. *PLoS ONE*, 11(1):e0147899. ISSN 19326203. doi:10.1371/journal.pone.0147899
- Milliken CE and May HD. 2007. Sustained generation of electricity by the spore-forming, Gram-positive, *Desulfitobacterium hafniense* strain DCB2. *Applied Microbiology and Biotechnology*, 73(5):1180–1189. ISSN 0175-7598. doi:10.1007/s00253-006-0564-6
- Min B, Cheng S and Logan BE. 2005. Electricity generation using membrane and salt bridge microbial fuel cells. *Water Research*, 39(9):1675–1686. ISSN 00431354. doi:10.1016/j.watres.2005.02.002
- Min B and Logan BE. 2004. Continuous electricity generation from domestic wastewater and organic substrates in a flat plate microbial fuel cell. *Environmental Science and Technology*, 38(21):5809–5814. ISSN 0013936X. doi:10.1021/es0491026

- Mohanakrishna G, Seelam JS, Vanbroekhoven K and Pant D. 2015. An enriched electroactive homoacetogenic biocathode for the microbial electrosynthesis of acetate through carbon dioxide reduction. *Faraday Discussions*, 183:445–462. ISSN 13645498. doi:10.1039/c5fd00041f
- Morris BE, Henneberger R, Huber H and Moissl-Eichinger C. 2013. Microbial syntrophy: Interaction for the common good. *FEMS Microbiology Reviews*, 37(3):384–406. ISSN 01686445. doi:10.1111/1574-6976.12019
- Müller AC and Bockmayr A. 2013. Fast thermodynamically constrained flux variability analysis. *Bioinformatics*, 29(7):903–909. ISSN 13674803. doi:10.1093/bioinformatics/btt059
- Muto A, Kotera M, Tokimatsu T, Nakagawa Z, Goto S and Kanehisa M. 2013. Modular architecture of metabolic pathways revealed by conserved sequences of reactions. *Journal of Chemical Information and Modeling*, 53(3):613–622. ISSN 15499596. doi:10.1021/ci3005379
- Nanchaiaiah Y, Mohan SV and Lens PN. 2016. Biological and Bioelectrochemical Recovery of Critical and Scarce Metals. *Trends in Biotechnology*, 34(2):137–155. ISSN 18793096. doi:10.1016/j.tibtech.2015.11.003
- Nealson KH and Rowe AR. 2016. Electromicrobiology: realities, grand challenges, goals and predictions. *Microbial Biotechnology*, 9(5):595–600. ISSN 17517915. doi:10.1111/1751-7915.12400
- Nevin KP, Woodard TL, Franks AE, Summers ZM and Lovley DR. 2010. Microbial electrosynthesis: feeding microbes electricity to convert carbon dioxide and water to multicarbon extracellular organic compounds. *mBio*, 1(2):e00103–10. ISSN 2150-7511. doi:10.1128/mBio.00103-10
- Nichols EM, Gallagher JJ, Liu C, Su Y, Resasco J, Yu Y, Sun Y, Yang P, Chang MC and Chang CJ. 2015. Hybrid bioinorganic approach to solar-to-chemical conversion. *Proceedings of the National Academy of Sciences*, 112(37):11461–11466. ISSN 0027-8424. doi:10.1073/pnas.1508075112
- Nimje VR, Chen CYCC, Chen HR, Chen CYCC, Tseng MJ, Cheng KC, Shih RC and Chang YF. 2012. A single-chamber microbial fuel cell without an air cathode. *International journal of molecular sciences*, 13(3):3933–48. ISSN 1422-0067. doi:10.3390/ijms13033933
- Nishio K, Hashimoto K and Watanabe K. 2010. Light/electricity conversion by a self-organized photosynthetic biofilm in a single-chamber reactor. *Applied Microbiology and Biotechnology*, 86(3):957–964. ISSN 01757598. doi:10.1007/s00253-009-2400-2
- Noguera DR, Brusseau GA, Rittmann BE and Stahl DA. 1998. A Unified Model Describing the Role of Hydrogen in the Growth of *Desulfovibrio vulgaris* under Different Environmental Conditions. *Biotechnology and Bioengineering*, 59(6)



- Oh SE and Logan BE. 2006. Proton exchange membrane and electrode surface areas as factors that affect power generation in microbial fuel cells. *Applied Microbiology and Biotechnology*, 70(2):162–169. ISSN 0175-7598. doi:10.1007/s00253-005-0066-y
- Okamoto A, Kalathil S, Deng X, Hashimoto K, Nakamura R and Nealsen KH. 2015. Cell-secreted Flavins Bound to Membrane Cytochromes Dictate Electron Transfer Reactions to Surfaces with Diverse Charge and pH. *Scientific Reports*, 4(1):5628. ISSN 2045-2322. doi:10.1038/srep05628
- Oscar TP, Spears JW and Shih J. 1987. Performance, Methanogenesis and Nitrogen Metabolism of Finishing Steers Fed Monensin and Nickel. *Journal of Animal Science*, 64(3):887–896. ISSN 0021-8812. doi:10.2527/jas1987.643887x
- Park JH, Kang HJ, Park KH and Park HD. 2018. Direct interspecies electron transfer via conductive materials: A perspective for anaerobic digestion applications. *Bioresource Technology*, 254. ISSN 18732976. doi:10.1016/j.biortech.2018.01.095
- Phelps TJ, Conrad R and Zeikus JG. 1985. Sulfate-dependent interspecies H<sub>2</sub> transfer between *Methanosarcina barkeri* and *Desulfovibrio vulgaris* during coculture metabolism of acetate or methanol. *Applied and Environmental Microbiology*, 50(3):589–594. ISSN 00992240. doi:0099-2240/85/090589-06\$02.00/0
- Pirbadian S, Barchinger SE, Leung KM, Byun HS, Jangir Y, Bouhenni RA, Reed SB, Romine MF, Saffarini DA, Shi L, Gorby YA, Golbeck JH and El-Naggar MY. 2014. *Shewanella oneidensis* MR-1 nanowires are outer membrane and periplasmic extensions of the extracellular electron transport components. *Proceedings of the National Academy of Sciences of the United States of America*, 111(35):12883–8. ISSN 1091-6490. doi:10.1073/pnas.1410551111
- Pisciotta JM, Zaybak Z, Call DF, Nam JY and Logan BE. 2012. Enrichment of microbial electrolysis cell biocathodes from sediment microbial fuel cell bioanodes. *Applied and Environmental Microbiology*, 78(15):5212–9. ISSN 1098-5336. doi:10.1128/AEM.00480-12
- Ponomarova O and Patil KR. 2015. Metabolic interactions in microbial communities: untangling the Gordian knot. *Current Opinion in Microbiology*, 27:37–44. ISSN 1369-5274. doi:10.1016/J.MIB.2015.06.014
- Potter MC. 1911. Electrical Effects Accompanying the Decomposition of Organic Compounds. *Proceedings of the Royal Society B: Biological Sciences*, 84(571):260–276. ISSN 0962-8452. doi:10.1098/rspb.1911.0073
- Price NC, Dwek RA, Ratcliffe RG and Wormald M. 2001. Principles and problems in physical chemistry for biochemists. Oxford University Press. ISBN 9780198792819

- Pruitt K, Brown G, Tatusova T and Maglott D. 2002. Chapter 18 The Reference Sequence (RefSeq) Database. In McEntyre J and Ostell J, (eds.) The NCBI Handbook [Internet], chapter 18. Bethesda (MD): National Center for Biotechnology Information (US)
- Purcell E. 2011. Electricity and Magnetism. September. Cambridge University Press Textbooks. ISBN 9781283315043
- Rabaey K. 2010. Bioelectrochemical Systems: a new approach towards environmental and industrial biotechnology. In Rabaey K, Angenent L, Schroder U and Keller J, (eds.) Bioelectrochemical systems: from extracellular electron transfer to biotechnological application, chapter 1, 488. London, UK: IWA Publishing. ISBN 184339233X
- Rabaey K, Angenent L, Schroeder U and Keller J, (eds.) . 2010. Electrochemical Techniques for The Analysis of Bioelectrochemical Systems. September 2018. IWA Publishing. ISBN 9781843392330
- Rabaey K, Clauwaert P, Aelterman P and Verstraete W. 2005. Tubular Microbial Fuel Cells for Efficient Electricity Generation. *Environmental Science & Technology*, 39(20). doi:10.1021/ES050986I
- Reguera G, Nevin KP, Nicoll JS, Covalla SF, Woodard TL and Lovley DR. 2006. Biofilm and nanowire production leads to increased current in *Geobacter sulfurreducens* fuel cells. *Applied and environmental microbiology*, 72(11):7345–8. ISSN 0099-2240. doi:10.1128/AEM.01444-06
- Rittmann BE. 2017. Ironies in Microbial Electrochemistry. *Journal of Environmental Engineering*, 143(5):03117001. ISSN 0733-9372. doi:10.1061/(ASCE)EE.1943-7870.0001202
- Rock PA. 1966. The standard oxidation potential of the ferrocyanide-ferricyanide electrode at 25°C and the entropy of ferrocyanide ion. *Journal of Physical Chemistry*, 70(2):576–580. ISSN 00223654. doi:10.1021/j100874a042
- Rodriguez-Concepcion M, Avalos J, Bonet ML, Boronat A, Gomez-Gomez L, Hornero-Mendez D, Limon MC, Meléndez-Martínez AJ, Olmedilla-Alonso B, Palou A, Ribot J, Rodrigo MJ, Zacarias L and Zhu C. 2018. A global perspective on carotenoids: Metabolism, biotechnology, and benefits for nutrition and health. *Progress in Lipid Research*, 70(February):62–93. ISSN 18732194. doi:10.1016/j.plipres.2018.04.004
- Roehm Kh. 2001. Electron Carriers : Proteins and Cofactors in Oxidative Phosphorylation. In Encyclopedia of Lifescience, 1–8. John Wiley & Sons, Ltd. ISBN 9780470015902. doi:10.1038/npg.els.0001373

- Ross DE, Flynn JM, Baron DB, Gralnick JA and Bond DR. 2011. Towards Electrosynthesis in *Shewanella*: Energetics of Reversing the Mtr Pathway for Reductive Metabolism. *PLoS ONE*, 6(2):e16649. ISSN 1932-6203. doi:10.1371/journal.pone.0016649
- Rotaru AE, Shrestha PM, Liu F, Markovaitė B, Chen S, Nevin KP and Lovley DR. 2014a. Direct interspecies electron transfer between *Geobacter metallireducens* and *Methanosarcina barkeri*. *Applied and environmental microbiology*, 80(15):4599–605. ISSN 1098-5336. doi:10.1128/AEM.00895-14
- Rotaru AE, Shrestha PM, Liu F, Shrestha M, Shrestha D, Embree M, Zengler K, Wardman C, Nevin KP and Lovley DR. 2014b. A new model for electron flow during anaerobic digestion: direct interspecies electron transfer to *Methanosaeta* for the reduction of carbon dioxide to methane. *Energy Environ. Sci.*, 7(1):408–415. ISSN 1754-5692. doi:10.1039/C3EE42189A
- Rousk J. 2016. Biomass or growth? How to measure soil food webs to understand structure and function. *Soil Biology and Biochemistry*, 102:45–47. ISSN 00380717. doi:10.1016/j.soilbio.2016.07.001
- Roy S, Schievano A and Pant D. 2015. Electro-stimulated microbial factory for value added product synthesis. *Bioresource Technology*, 213:129–139. ISSN 18732976. doi:10.1016/j.biortech.2016.03.052
- Rozendal R, Xe, A, Jeremiasse AW, Hamelers HVM and Buisman CJN. 2008. Hydrogen Production with a Microbial Biocathode. *Environmental Science & Technology*, 42(2):629–634. ISSN 0013-936x
- Rusling JF. 2018. Developing Microfluidic Sensing Devices Using 3D Printing. *ACS Sensors*, 3(3):522–526. ISSN 2379-3694. doi:10.1021/acssensors.8b00079
- Sadhukhan J, Lloyd JR, Scott K, Premier GC, Yu EH, Curtis T and Head IM. 2016. A critical review of integration analysis of microbial electrosynthesis (MES) systems with waste biorefineries for the production of biofuel and chemical from reuse of CO<sub>2</sub>. *Renewable and Sustainable Energy Reviews*, 56:116–132. ISSN 1364-0321. doi:10.1016/J.RSER.2015.11.015
- Santoro C, Arbizzani C, Erable B and Ieropoulos IA. 2017. Microbial fuel cells: From fundamentals to applications. A review. *Journal of Power Sources*, 356:225–244. ISSN 0378-7753. doi:10.1016/J.JPOWSOUR.2017.03.109
- Santoro C, Ieropoulos IA, Greenman J, Cristiani P, Vadas T, Mackay A and Li B. 2013. Current generation in membraneless single chamber microbial fuel cells (MFCs) treating urine. *Journal of Power Sources*, 238:190–196. ISSN 03787753. doi:10.1016/j.jpowsour.2013.03.095

- Sañudo-Wilhelmy SA, Gómez-Consarnau L, Suffridge C and Webb EA. 2014. The Role of B Vitamins in Marine Biogeochemistry. *Annual Review of Marine Science*, 6(1):339–367. ISSN 1941-1405. doi:10.1146/annurev-marine-120710-100912
- Sasidharan K, Martinez-Vernon AS, Chen J, Fu T and Soyer O. 2018. A low-cost DIY device for high resolution, continuous measurement of microbial growth dynamics. *bioRxiv*, 407742. doi:10.1101/407742
- Schievano A, Pepé Sciarria T, Vanbroekhoven K, De Wever H, Puig S, Andersen SJ, Rabaey K and Pant D. 2016. Electro-Fermentation – Merging Electrochemistry with Fermentation in Industrial Applications. *Trends in Biotechnology*, 34(11):866–878. ISSN 18793096. doi:10.1016/j.tibtech.2016.04.007
- Schink B. 1997. Energetics of syntrophic cooperation in methanogenic degradation. *Microbiology and molecular biology reviews : MMBR*, 61(2):262–280. ISSN 1092-2172. doi:1092-2172/97/\$04.0010
- Schmitz S, Nies S, Wierckx N, Blank LM and Rosenbaum MA. 2015. Engineering mediator-based electroactivity in the obligate aerobic bacterium *Pseudomonas putida* KT2440. *Frontiers in Microbiology*, 6:284. ISSN 1664-302X. doi:10.3389/fmicb.2015.00284
- Schoepp-Cothenet B, Van Lis R, Atteia A, Baymann F, Capowiez L, Ducluzeau AL, Duval S, Ten Brink F, Russell MJ and Nitschke W. 2013. On the universal core of bioenergetics. *Biochimica et Biophysica Acta - Bioenergetics*, 1827(2):79–93. ISSN 00052728. doi:10.1016/j.bbabbio.2012.09.005
- Scholz F, (ed.) . 2010. Electroanalytical Methods. Springer Berlin Heidelberg. ISBN 9783642029141. doi:10.1007/978-3-642-02915-8
- Sharma M, Aryal N, Sarma PM, Vanbroekhoven K, Lal B, Benetton XD and Pant D. 2013. Bioelectrocatalyzed reduction of acetic and butyric acids via direct electron transfer using a mixed culture of sulfate-reducers drives electrosynthesis of alcohols and acetone. *Chemical Communications*, 49(58):6495–6497. ISSN 1364548X. doi:10.1039/c3cc42570c
- Shemfe M, Gadkari S, Yu E, Rasul S, Scott K, Head IM, Gu S and Sadhukhan J. 2018. Life cycle, techno-economic and dynamic simulation assessment of bioelectrochemical systems: A case of formic acid synthesis. *Bioresource Technology*, 255(November 2017):39–49. ISSN 18732976. doi:10.1016/j.biortech.2018.01.071

- Shi L, Dong H, Reguera G, Beyenal H, Lu A, Liu J, Yu HQQ and Fredrickson JK. 2016. Extra-cellular electron transfer mechanisms between microorganisms and minerals. *Nature Reviews Microbiology*, 14(10):651–662. ISSN 1740-1526. doi:10.1038/nrmicro.2016.93
- Shin HJ, Jung KA, Nam CW and Park JM. 2017. A genetic approach for microbial electrosynthesis system as biocommodities production platform. *Bioresource Technology*, 245:1421–1429. ISSN 18732976. doi:10.1016/j.biortech.2017.05.077
- Singh S, Dwivedi C and Pandey A. 2016. Electricity generation in membrane-less single chambered microbial fuel cell. In 2016 International Conference on Control, Computing, Communication and Materials (ICCCCM), 1–4. IEEE. ISBN 978-1-4673-9084-2. doi:10.1109/ICCCCM.2016.7918216
- Soh KC and Hatzimanikatis V. 2010. Network thermodynamics in the post-genomic era. doi:10.1016/j.mib.2010.03.001
- Stams AJM and Plugge CM. 2009. Electron transfer in syntrophic communities of anaerobic bacteria and archaea. *Nature Reviews Microbiology*, 7(8):568–577. ISSN 17401526. doi:10.1038/nrmicro2166
- Stratford JP, Beecroft NJ, Slade RC, Grüning A and Avignone-Rossa C. 2014. Anodic microbial community diversity as a predictor of the power output of microbial fuel cells. *Bioresource Technology*, 156:84–91. ISSN 0960-8524. doi:10.1016/J.BIORTECH.2014.01.041
- Summers ZM, Fogarty HE, Leang C, Franks AE, Malvankar NS and Lovley DR. 2010. Direct exchange of electrons within aggregates of an evolved syntrophic coculture of anaerobic bacteria. *Science*, 330(6009):1413–5. ISSN 1095-9203. doi:10.1126/science.1196526
- Sun D, Wang A, Cheng S, Yates M and Logan BE. 2014. *Geobacter anodireducens* sp. nov., an exoelectrogenic microbe in bioelectrochemical systems. *International Journal of Systematic and Evolutionary Microbiology*, 64(Pt 10):3485–3491. ISSN 1466-5026. doi:10.1099/ij.s.0.061598-0
- Sund CJ, McMasters S, Crittenden SR, Harrell LE and Sumner JJ. 2007. Effect of electron mediators on current generation and fermentation in a microbial fuel cell. *Applied Microbiology and Biotechnology*, 76(3):561–568. ISSN 01757598. doi:10.1007/s00253-007-1038-1
- Sung J, Kim S, Cabatbat JJT, Jang S, Jin YS, Jung GY, Chia N and Kim PJ. 2017. Global metabolic interaction network of the human gut microbiota for context-specific community-scale analysis. *Nature Communications*, 8:15393. ISSN 2041-1723. doi:10.1038/ncomms15393
- Tatusov RL, Galperin MY, Natale DA and Koonin EV. 2000. The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic acids research*, 28(1):33–36

- Tenenbaum D. 2018. KEGGREST: Client-side REST access to KEGG. *R package 1.20.0*. doi: 10.18129/B9.bioc.KEGGREST
- Thauer RK, Jungermann K and Decker K. 1977. Energy conservation in chemotrophic anaerobic bacteria. *Bacteriological reviews*, 41(1):100–80. ISSN 0005-3678
- Tian X, Zhao F, You L, Wu X, Zheng Z, Wu R, Jiang Y and Sun S. 2017. Interaction between in vivo bioluminescence and extracellular electron transfer in *Shewanella woodyi* via charge and discharge. *Physical Chemistry Chemical Physics*, 19(3):1746–1750. ISSN 1463-9076. doi: 10.1039/C6CP07595A
- Torres CI, Marcus AK, Parameswaran P and Rittmann BE. 2008. Kinetic experiments for evaluating the nernst-monod model for anode-respiring bacteria (ARB) in a biofilm anode. *Environmental Science and Technology*, 42(17):6593–6597. ISSN 0013936X. doi:10.1021/es800970w
- Trapero JR, Horcajada L, Linares JJ and Lobato J. 2017. Is microbial fuel cell technology ready? An economic answer towards industrial commercialization. *Applied Energy*, 185:698–707. ISSN 03062619. doi:10.1016/j.apenergy.2016.10.109
- Tremblay PL and Zhang T. 2015. Electrifying microbes for the production of chemicals. *Frontiers in Microbiology*, 6(MAR):201. ISSN 1664302X. doi:10.3389/fmicb.2015.00201
- Upadhyay SK, (ed.) . 2006. Chemical Kinetics and Reaction Dynamics. Dordrecht: Springer Netherlands. ISBN 978-1-4020-4546-2. doi:10.1007/978-1-4020-4547-9
- Venkata Mohan S, Veer Raghavulu S and Sarma P. 2008. Influence of anodic biofilm growth on bioelectricity production in single chambered mediatorless microbial fuel cell using mixed anaerobic consortia. *Biosensors and Bioelectronics*, 24(1):41–47. ISSN 09565663. doi:10.1016/j.bios.2008.03.010
- Walker CB, He Z, Yang ZK, Ringbauer JA, He Q, Zhou J, Voordouw G, Wall JD, Arkin AP, Hazen TC, Stolyar S and Stahl DA. 2009. The electron transfer system of syntrophically grown *Desulfovibrio vulgaris*. *Journal of Bacteriology*, 191(18):5793–5801. ISSN 00219193. doi:10.1128/JB.00356-09
- Wang L, Liu L and Yang F. 2018. Efficient gas phase VOC removal and electricity generation in an integrated bio-photo-electro-catalytic reactor with bio-anode and TiO<sub>2</sub> photo-electro-catalytic air cathode. *Bioresource Technology*. ISSN 09608524. doi:10.1016/j.biortech.2018.09.041
- Wang VB, Chua SL, Cao B, Seviour T, Nesatyy VJ, Marsili E, Kjelleberg S, Givskov M, Tolker-Nielsen T, Song H, Loo JSC and Yang L. 2013. Engineering PQS Biosynthesis Pathway for

- Enhancement of Bioelectricity Production in *Pseudomonas aeruginosa* Microbial Fuel Cells. *PLoS ONE*, 8(5):e63129. ISSN 1932-6203. doi:10.1371/journal.pone.0063129
- Watanabe K, Manefield M, Lee M and Kouzuma A. 2009. Electron shuttles in biotechnology. *Current Opinion in Biotechnology*, 20(6):633–641. ISSN 09581669. doi:10.1016/j.copbio.2009.09.006
- Wei J, Liang P and Huang X. 2011. Recent progress in electrodes for microbial fuel cells. *Biore-source Technology*, 102(20):9335–9344. ISSN 0960-8524. doi:10.1016/J.BIORTECH.2011.07.019
- Whitman WB, Rainey F, Kämpfer P, Trujillo M, Chun J, DeVos P, Hedlund B and Dedysh S, (eds.) . 2015. Bergey’s manual of systematics of Archaea and Bacteria. Chichester, UK: John Wiley & Sons, Ltd. ISBN 9781118960608. doi:10.1002/9781118960608
- Wodke J, Puchalka J, Lluch-Senar M, Marcos J, Yus E, Godinho M, Gutierrez-Gallego R, Martins dos Santos VA, Serrano L, Klipp E and Maier T. 2014. Dissecting the energy metabolism in *Mycoplasma pneumoniae* through genome-scale metabolic modeling. *Molecular Systems Biology*, 9(1):653–653. ISSN 1744-4292. doi:10.1038/msb.2013.6
- Wu Y, Liu T, Li X and Li F. 2014. Exogenous electron shuttle-mediated extracellular electron transfer of *Shewanella putrefaciens* 200: Electrochemical parameters and thermodynamics. *Environmental Science and Technology*, 48(16):9306–9314. ISSN 15205851. doi:10.1021/es5017312
- Xu A, Dolfing J, Curtis TP, Montague G and Martin E. 2011. Maintenance affects the stability of a two-tiered microbial ‘food chain’? *Journal of Theoretical Biology*, 276(1):35–41. ISSN 00225193. doi:10.1016/j.jtbi.2011.01.026
- Xu S, Jangir Y and El-Naggar MY. 2016. Disentangling the roles of free and cytochrome-bound flavins in extracellular electron transport from *Shewanella oneidensis* MR-1. *Electrochimica Acta*, 198:49–55. ISSN 00134686. doi:10.1016/j.electacta.2016.03.074
- Yang X, Ma X, Wang K, Wu D, Lei Z and Feng C. 2016a. Eighteen-month assessment of 3D graphene oxide aerogel-modified 3D graphite fiber brush electrode as a high-performance microbial fuel cell anode. *Electrochimica Acta*, 210:846–853. ISSN 0013-4686. doi:10.1016/J.ELECTACTA.2016.05.215
- Yang YJ, Sheu BS, Yang YJ and Sheu BS. 2016b. Metabolic Interaction of *Helicobacter pylori* Infection and Gut Microbiota. *Microorganisms*, 4(1):15. ISSN 2076-2607. doi:10.3390/microorganisms4010015
- Yeagle P. 2016. The membranes of cells. Academic Press, third edit edition. ISBN 9780128000472

- Yong XY, Feng J, Chen YL, Shi DY, Xu YS, Zhou J, Wang SY, Xu L, Yong YC, Sun YM, Shi CL, OuYang PK and Zheng T. 2014. Enhancement of bioelectricity generation by cofactor manipulation in microbial fuel cell. *Biosensors and Bioelectronics*, 56:19–25. ISSN 0956-5663. doi:10.1016/J.BIOS.2013.12.058
- Yong XY, Yan ZY, Shen HB, Zhou J, Wu XY, Zhang LJ, Zheng T, Jiang M, Wei P, Jia HH and Yong YC. 2017. An integrated aerobic-anaerobic strategy for performance enhancement of *Pseudomonas aeruginosa*-inoculated microbial fuel cell. *Bioresource Technology*, 241:1191–1196. ISSN 0960-8524. doi:10.1016/J.BIORTECH.2017.06.050
- Yu EH. 2015. Resource Recovery with Microbial Electrochemical Systems. *Microbial Electrochemical and Fuel Cells: Fundamentals and Applications*, 321–339. doi:10.1016/B978-1-78242-375-1.00010-1
- Yu G, Wang LG, Han Y and He QY. 2012. clusterProfiler: an R Package for Comparing Biological Themes Among Gene Clusters. *OMICS: A Journal of Integrative Biology*, 16(5):284–287. ISSN 1536-2310. doi:10.1089/omi.2011.0118
- Yu Z, Leng X, Zhao S, Ji J, Zhou T, Khan A, Kakde A, Liu P and Li X. 2018. A review on the applications of microbial electrolysis cells in anaerobic digestion. *Bioresource Technology*, 255(November 2017):340–348. ISSN 18732976. doi:10.1016/j.biortech.2018.02.003
- Zaybak Z, Pisciotta JM, Tokash JC and Logan BE. 2013. Enhanced start-up of anaerobic facultatively autotrophic biocathodes in bioelectrochemical systems. *Journal of Biotechnology*, 168(4):478–485. ISSN 01681656. doi:10.1016/j.jbiotec.2013.10.001
- Zehnder AJ and Wuhrmann K. 1976. Titanium (III) citrate as a nontoxic oxidation-reduction buffering system for the culture of obligate anaerobes. *Science (New York, N.Y.)*, 194(4270):1165–6. ISSN 0036-8075
- Zhang JD and Wiemann S. 2009. KEGGgraph: A graph approach to KEGG PATHWAY in R and bioconductor. *Bioinformatics*, 25(11):1470–1471. ISSN 13674803. doi:10.1093/bioinformatics/btp167
- Zhang X, Liu H, Wang J, Ren G, Xie B, Liu H, Zhu Y and Jiang L. 2015. Facilitated extracellular electron transfer of *Shewanella loihica* PV-4 by antimony-doped tin oxide nanoparticles as active microelectrodes. *Nanoscale*, 7(44):18763–9. ISSN 2040-3372. doi:10.1039/c5nr04765j
- Zhao S, Guo Y and Shyr Y. 2017a. KEGGprofile: An annotation and visualization package for multi-types and multi-groups expression data in KEGG pathway. *R package version 1.22.0*



- Zhao Z, Zhang Y, Li Y, Dang Y, Zhu T and Quan X. 2017b. Potentially shifting from interspecies hydrogen transfer to direct interspecies electron transfer for syntrophic metabolism to resist acidic impact with conductive carbon cloth. *Chemical Engineering Journal*, 313:10–18. ISSN 13858947. doi:10.1016/j.cej.2016.11.149
- Zhu G, Yang Y, Liu J, Liu F, Lu A and He W. 2017. Enhanced photocurrent production by the synergy of hematite nanowire-arrayed photoanode and bioengineered *Shewanella oneidensis* MR-1. *Biosensors and Bioelectronics*, 94:227–234. ISSN 0956-5663. doi:10.1016/J.BIOS.2017.03.006
- Zhuang L, Zhou S, Li Y and Yuan Y. 2010. Enhanced performance of air-cathode two-chamber microbial fuel cells with high-pH anode and low-pH cathode. *Bioresource Technology*, 101(10):3514–3519. ISSN 0960-8524. doi:10.1016/J.BIORTECH.2009.12.105
- Zuo Y, Xing D, Regan JM and Logan BE. 2008. Isolation of the exoelectrogenic bacterium *Ochrobactrum anthropi* YZ-1 by using a U-tube microbial fuel cell. *Applied and Environmental Microbiology*, 74(10):3130–3137. ISSN 00992240. doi:10.1128/AEM.02732-07

# Appendix A

## Thermodynamics and metabolism

### A.1 Thermodynamics and metabolism

Tables A.1 and A.2 are a tabular representation of Figures 1.2 and 1.3, respectively.

**Table A.1** Reduction potential chart – a thermodynamic view of metabolic redox reactions. The reduction potential of some reduction half reactions that occur in microbial metabolism are listed according to their reduction potential ( $E_h^o$ ). For simplicity, some multi-step reactions in metabolism, such as pyruvate to glucose, are represented as a single step reaction. In these cases, the reduction potential of the overall reaction is given (system property, Alberty, 2003). The half reactions have been classified by ‘type’ indicating whether they involve organic compounds ( $C_xH_yO_z$ ), electron carriers (EC) or TEAs. Data obtained from Thauer et al. (1977) and Alberty (2001). ox, oxidised; rd, reduced; FAD, flavin adenine dinucleotide.

	Reaction	$E_{reduction}^o$ (mV)	Type
1	2 Pyruvate $\rightarrow$ Glucose	-704.46	$C_xH_yO_z$
2	Acetate + $HCO_3 \rightarrow$ Pyruvate	-659.12	$C_xH_yO_z$
3	Acetate $\rightarrow$ Acetaldehyde	-580.86	$C_xH_yO_z$
4	Ferredoxin ox $\rightarrow$ Ferredoxin rd	-420.00	EC
5	Acetyl-CoA $\rightarrow$ Acetaldehyde	-420.00	$C_xH_yO_z$
6	Acetate $\rightarrow$ Ethanol	-389.13	$C_xH_yO_z$
7	Flavodoxin ox $\rightarrow$ Flavodoxin rd	-371.00	EC
8	$NAD^+ \rightarrow$ NADH	-320.00	EC
9	Cytochrome c ox $\rightarrow$ Cytochrome c rd	-290.00	EC
10	$CO_2 \rightarrow$ Methane	-244.00	TEA
11	$FAD \rightarrow$ $FADH_2$	-220.00	EC
12	Sulfate $\rightarrow$ Sulfide	-216.82	TEA
13	Sulfate $\rightarrow$ Sulfur	-199.29	TEA
14	Acetaldehyde $\rightarrow$ Ethanol	-197.39	$C_xH_yO_z$
15	Pyruvate $\rightarrow$ Lactate	-190.65	$C_xH_yO_z$
16	Rubredoxin ox $\rightarrow$ Rubredoxin rd	-57.00	EC
17	$2 H^+ \rightarrow$ $H_2$	0.00	TEA
18	Fumarate $\rightarrow$ Succinate	32.70	$C_xH_yO_z$
19	Ubiquinone $\rightarrow$ Ubiquinol	113.00	EC
20	$2 NO_3 \rightarrow$ $N_2$	747.32	TEA
21	$O_2 + 2 H_2 \rightarrow 2 H_2O$	815.46	TEA

**Table A.2** Three types of microorganism were roughly mapped onto the reduction potential chart based on known metabolic processes they carry out, each represented by the greyed area parallel to the half reactions. Methanogens are known to reduce  $\text{CO}_2$  to  $\text{CH}_4$ . Sulphate-reducing bacteria are known to reduce sulphate to either sulphide or sulphur, as their name suggests. Finally, denitrifying bacteria reduce  $\text{NO}_3$  to  $\text{N}_2$ . These processes refer to the “last” reactions that the organisms can carry out and hence have determined the bottom range for the species assignment. However, the question of the upper bound that should be assigned to each remains open, as it would depend on their genetic capacity. Here, the assumption that all microorganisms can oxidise glucose to pyruvate has been made. The arrows show the range of possible half reactions for each species, limiting the maximum energy (in the form of potential difference) available to them. Therefore, methanogens can harvest much less energy than sulphate-reducing or denitrifying bacteria. Data obtained from Thauer et al. (1977) and Alberty (2001).

	Reaction	$E_{\text{reduction}}^{\circ}$ (mV)	Methanogen	Sulphate-reducer	Denitrifying
1	2 Pyruvate $\rightarrow$ Glucose	-704.46	↑	↑	↑
2	Acetate + $\text{HCO}_3 \rightarrow$ Pyruvate	-659.12			
3	Acetate $\rightarrow$ Acetaldehyde	-580.86			
4	Ferredoxin ox $\rightarrow$ Ferredoxin rd	-420.00			
5	Acetyl-CoA $\rightarrow$ Acetaldehyde	-420.00			
6	Acetate $\rightarrow$ Ethanol	-389.13			
7	Flavodoxin ox $\rightarrow$ Flavodoxin rd	-371.00			
8	$\text{NAD}^+ \rightarrow$ NADH	-320.00			
9	Cytochrome c ox $\rightarrow$ Cytochrome c rd	-290.00			
10	$\text{CO}_2 \rightarrow$ Methane	-244.00	↓		
11	$\text{FAD} \rightarrow$ $\text{FADH}_2$	-220.00			
12	Sulfate $\rightarrow$ Sulfide	-216.82			
13	Sulfate $\rightarrow$ Sulfur	-199.29		↓	
14	Acetaldehyde $\rightarrow$ Ethanol	-197.39			
15	Pyruvate $\rightarrow$ Lactate	-190.65			
16	Rubredoxin ox $\rightarrow$ Rubredoxin rd	-57.00			
17	$2 \text{H}^+ \rightarrow$ $\text{H}_2$	0.00			
18	Fumarate $\rightarrow$ Succinate	32.70			
19	Ubiquinone $\rightarrow$ Ubiquinol	113.00			
20	$2 \text{NO}_3 \rightarrow$ $\text{N}_2$	747.32			↓
21	$\text{O}_2 + 2 \text{H}_2 \rightarrow 2 \text{H}_2\text{O}$	815.46			

# Appendix B

## Electrochemical platform

### B.1 Electrochemical cell measurements

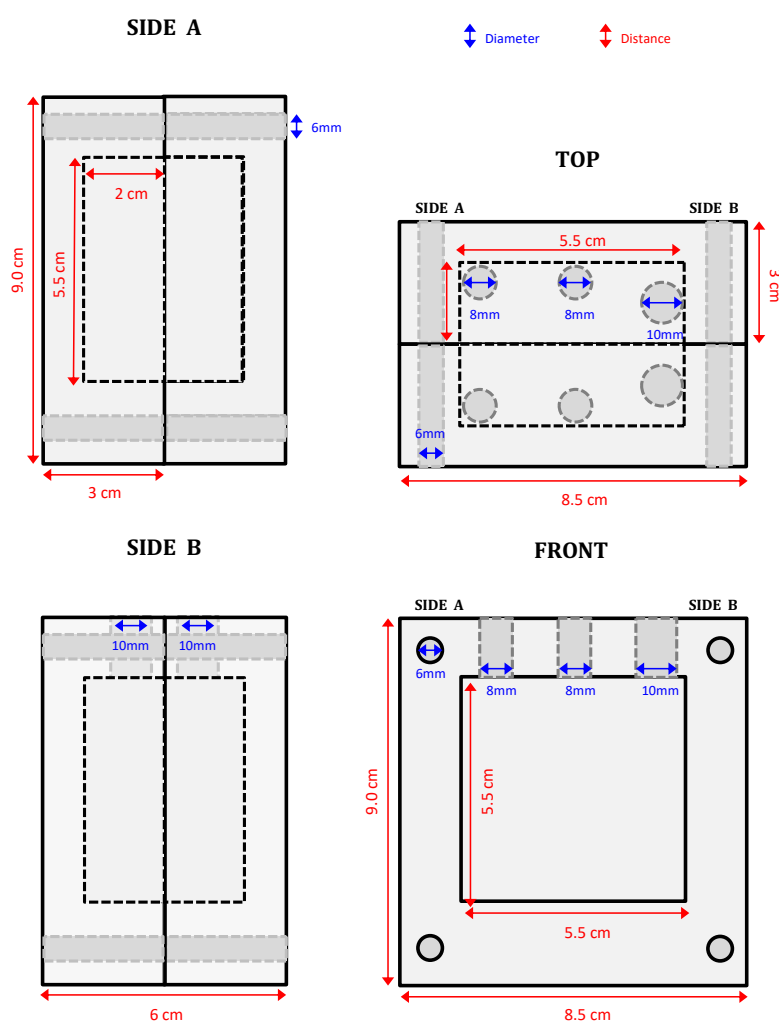


Figure B.1 Electrochemical cell measurements.

### B.2 OpenSCAD code for 3D printed objects

#### B.2.1 Rubber gasket

**Code B.1** OpenSCAD code to produce 3D printed stencil to cut the rubber gaskets to required dimensions

```
1 //ELECTRODE STENSIL TO CUT TO SHAPE.
2 //design by AMV & KS
3 //01.12.2017, Warwick, UK
4
5 //// PARAMETERS - All units in mm
6
7 //STENSIL MAIN FRAME
8 cubeW = 85;
9 cubeL = 90;
10 cubeH = 3;
11
12 //CUT INSIDE SQUARE
13 cubeCut_W = 55;
14 cubeCut_L = 55;
15 cubeCut_H = cubeH+5;
16
17 //BOLT
18 bolt_R = 3.6;
19 bolt_H = cubeH+5;
20
21 ////MAKE PRINT
22 difference() {
23     //CUBE
24     translate([0,0,0]) color("black") cube([cubeW,cubeL,cubeH], false, $fn=60);
25     //INSIDE FRAME
26     translate([15,20,-1]) color("black") cube([cubeCut_W,cubeCut_L,cubeCut_H], false, $fn=60);
27     //BOLTS
28     translate([-cubeW/2+10,-cubeL/2+10,0]) color("red") cylinder(h = bolt_H, r1 = bolt_R, r2 = bolt_R, center = false, $fn=60);
29     //BOLTS - bottom left
30     translate([-cubeW/2+10,-cubeL/2+10,-1]) color("red") cylinder(h = bolt_H, r1 = bolt_R, r2 = bolt_R, center = false, $fn=60);
31
32     //BOLTS - bottom left
33     translate([10,10,-1]) color("green") cylinder(h = bolt_H, r1 = bolt_R, r2 = bolt_R, center = false, $fn=60);
34     //BOLTS - bottom right
35     translate([cubeW-10,10,-1]) color("green") cylinder(h = bolt_H, r1 = bolt_R, r2 = bolt_R, center = false, $fn=60);
36     //BOLTS - top left
37     translate([10,cubeL-10,-1]) color("green") cylinder(h = bolt_H, r1 = bolt_R, r2 = bolt_R, center = false, $fn=60);
38     //BOLTS - top left
39     translate([cubeW-10,cubeL-10,-1]) color("green") cylinder(h = bolt_H, r1 = bolt_R, r2 = bolt_R, center = false, $fn=60);
40     {//// CUT SLITS HORIZONTAL
41         // bottom left
42         translate([13,20,-1]) color("black") cube([5,1,cubeCut_H], false, $fn=60);
43         // bottom right
44         translate([cubeW-15-2,20,-1]) color("black") cube([4,1,cubeCut_H], false, $fn=60);
45         // top left
46         translate([13,cubeL-16,-1]) color("black") cube([5,1,cubeCut_H], false, $fn=60);
47         // bottom right
48         translate([cubeW-15-2,cubeL-16,-1]) color("black") cube([4,1,cubeCut_H], false, $fn=60);
49     }
```

```

50
51 {//// CUT SLITS VERTICAL
52 // bottom left
53 translate ([15,18,-1]) color ("black") cube ([1,5,cubeCut_H], false, $fn=60);
54 // bottom right
55 translate ([cubeW-16,18,-1]) color ("black") cube ([1,5,cubeCut_H], false, $fn=60);
56 // top left
57 translate ([15,cubeL-18,-1]) color ("black") cube ([1,5,cubeCut_H], false, $fn=60);
58 // top right
59 translate ([cubeW-16,cubeL-18,-1]) color ("black") cube ([1,5,cubeCut_H], false, $fn
=60);
60 }
61 }

```

## B.2.2 Carbon fibre twill preparation

**Code B.2** OpenSCAD code to produce 3D printed frame stencil to place epoxy resin on carbon fibre square to produce electrode

```

1 //ELECTRODE STENSIL TO PLACE EPOXY USING A FRAME. EPOXY IS TO BE DRAWN WITHIN THE
  FRAME.
2 //design by AMV & KS
3 //01.12.2017, Warwick, UK
4
5 //// PARAMETERS - All units in mm
6
7 //STENSIL SQUARE
8 cubeW = 40;
9 cubeL = 40;
10 cubeH = 3;
11 frame_thickness = 2;
12
13 ////MAKE PRINT
14 difference(){
15 //CUBE
16 translate ([0,0,cubeH/2]) color ("yellow") cube ([cubeW+frame_thickness,cubeL+
frame_thickness,cubeH], true, $fn=60);
17 //CUBE
18 translate ([0,0,cubeH/2]) color ("yellow") cube ([cubeW,cubeL,cubeH+2], true, $fn
=60);
19 } // "TOP difference" ends

```

## B.2.3 Rubber stopper modification

**Code B.3** OpenSCAD code to produce 3D printed holder to pierce rubber bun with needle

```

1 //HOLDER TO PIERCE STOPPER WITH NEEDLE
2 //design by AMV & KS
3 //17.11.2017, Warwick, UK
4
5 // PARAMETERS - All units in mm
6
7 // STOPPER
8 stopper_R_top = 8.5/2;
9 stopper_R_bottom = 7/2;
10 stopper_H = 14;
11

```

```

12 //NEEDLE
13 needle_R = 1.5/2; //0.3 for blue needle
14 needle_H = 35;
15 needle_protusion_H = 5; // room for the needle to come through other side
16 needle_cannal_H = needle_H - stopper_H - needle_protusion_H;
17
18 //SUPPORT
19 cubeW = 15;
20 cubeL = 15;
21 cubeH = stopper_H+needle_cannal_H;
22
23 // MAKE PRINT
24 difference(){
25     //CUBE
26     translate([0,0,cubeH/2]) color("black") cube([cubeW,cubeL,cubeH],true,$fn=60);
27     //NEEDLE
28     translate([0,0,-1]) color("pink") cylinder(h = needle_H, r1 = needle_R, r2 =
        needle_R, center = false,$fn=60);
29     //STOPPER
30     translate([0,0,needle_cannal_H]) color("green") cylinder(h = stopper_H+1, r1 =
        stopper_R_bottom, r2 = stopper_R_top, center = false,$fn=60);
31 } // "TOP difference" ends

```

**Code B.4** OpenSCAD code to produce 3D printed holder support to pierce rubber bun with needle

```

1 //SUPPORT FOR HOLDER TO PIERCE STOPPER WITH NEEDLE
2 //design by AMV & KS
3 //17.11.2017, Warwick, UK
4
5 //// PARAMETERS - All units in mm
6
7 // STOPPER
8 stopper_R_top = 9/2;
9
10 //STOPPER LIP
11 stopper_lip_R = stopper_R_top/1.5;
12 stopper_lip_H = 3;
13
14 //NEEDLE
15 needle_protusion_H = 6;
16
17 //CUBE LIP
18 cube_lip_W = 15.5;
19 cube_lip_L = 15.5;
20 cube_lip_H = 6; // How much more should this support and the stopper holder should
    overlap
21
22 //SUPPORT
23 cubeW = cube_lip_W+10;
24 cubeL = cube_lip_L+10;
25 cubeH = needle_protusion_H + stopper_lip_H + cube_lip_H;
26
27 //CUBES ARE PLACED HALF THEIR HEIGHT down Z, but centered in X and Y if centre =
    true
28 //CYLINDERS ARE PLACED AT Z = 0 and centered in X and Y if centre = true
29
30 //// MAKE PRINT
31 difference(){
32     //CUBE

```

```

33     translate([0,0,cubeH/2]) color("yellow") cube([cubeW,cubeL,cubeH],true,$fn=60);
34     //CUBE LIP
35     translate([0,0,needle_protusion_H+stopper_lip_H+cubeH/2]) color("black") cube([
cube_lip_W,cube_lip_L,cubeH],true,$fn=60);
36     //STOPPER LIP
37     translate([0,0,needle_protusion_H]) color("red") cylinder(h=cubeH,r1=
stopper_R_top,r2=stopper_R_top,center=false,$fn=60);
38     //NEEDLE CANNAL
39     translate([0,0,-1]) color("green") cylinder(h=cubeH,r1=stopper_lip_R,r2=
stopper_lip_R,center=false,$fn=60);
40 } // "TOP difference" ends

```

## B.2.4 Electrode support

**Code B.5** OpenSCAD code to produce 3D printed electrode support to ensure the support of the working electrode in half-cell A and its separation from the counter electrode

```

1 //ELECTRODE SUPPORT TO SEPARATE AUXILIARY ELECTRODE FROM WORKING ELECTRODE
2 //design by AMV & KS
3 //06.04.2018, Warwick, UK
4
5 //// PARAMETERS - All units in mm
6
7 //EXTERNAL SQUARE
8 cubeW = 52;
9 cubeL = cubeW;
10 cubeH = 5;
11 frame_thickness = 2;
12
13 //CROSS SUPPORT
14 cubeW_s = sqrt((cubeW-frame_thickness)*(cubeW-frame_thickness)
15               +(cubeL-frame_thickness)*(cubeL-frame_thickness));
16 cubeL_s = frame_thickness;
17 cubeH_s = 5;
18
19 //WORKING ELECTRODE SUPPORT
20 cubeW_w = 40;
21 cubeL_w = cubeW_w;
22 cubeH_w = 2;
23
24 //ELECTRODE (FOR VISUALISATION PURPOSES)
25 cubeW_e = 40;
26 cubeL_e = 40;
27 cubeH_e = 1;
28
29 ////MAKE PRINT
30 difference(){ // TO ALLOW CROSS SECTION VIEW
31 union(){ // TO VISUALISE ELECTRODE
32 difference(){
33     union(){
34         //EXTERNAL FRAME
35         difference(){
36             translate([0,0,cubeH/2]) color("black") cube([cubeW,cubeL,cubeH],true,
$fn=60);
37             translate([0,0,cubeH/2]) color("black") cube([cubeW-frame_thickness*2,
cubeL-frame_thickness*2,cubeH+2],true,$fn=60);
38         }
39         //CROSS FOR SUPPORT

```



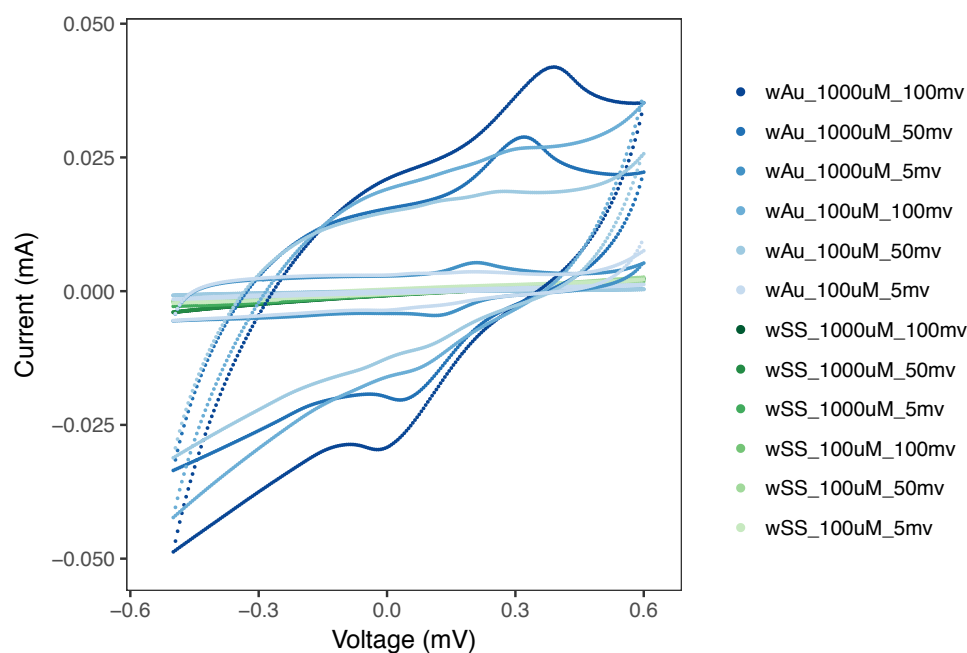
```

40     rotate([0,0,45]) translate([0,0,cubeH/2]) color("black") cube([cubeW_s,
cubeL_s,cubeH],true,$fn=60);
41     rotate([0,0,45]) translate([0,0,cubeH/2]) color("black") cube([cubeL_s,
cubeW_s,cubeH],true,$fn=60);
42 }
43 // INTERNAL CUT FOR ELECTROEDE
44 translate([0,0,frame_thickness/2+cubeH-frame_thickness]) color("red") cube([
cubeW_w,cubeL_w,frame_thickness+1],true,$fn=60);
45 }// TOP "DIFFERENCE"
46 }// TOP "UNION"
47 }// TOP "DIFFERENCE"

```

### B.3 Determining the material used for WE and CE connections

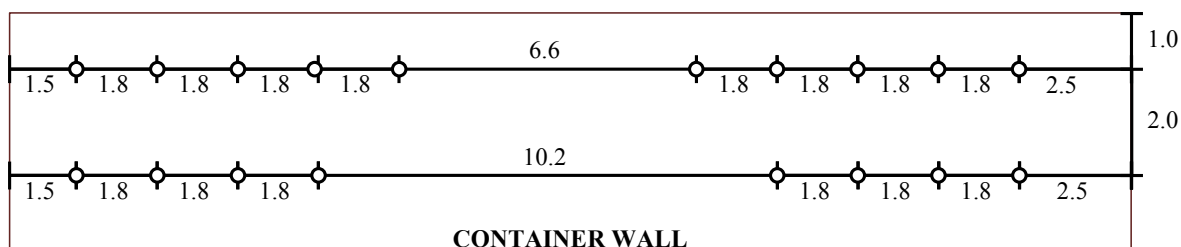
A performance comparison was carried out to determine what was the adequate wire material to produce an appropriate electrode connection. Potassium ferricyanide ( $\text{K}_3[\text{Fe}(\text{CN})_6]$ ) is a redox-active compound with a well defined cyclic voltammetry (CV) curve (Rock, 1966) (see Section 3.4.10 for details). CV was used to evaluate the ability of the different electrode connections to sense the redox reaction of a 100  $\mu\text{M}$   $\text{K}_3[\text{Fe}(\text{CN})_6]$  solution using a GAMRY potentiostat (see Section 2.1). Two CFTS coated with 10 and 25 nm of gold were used as the working and counter electrodes, respectively. The former was connected to the working and working sense leads of the GAMRY potentiostat, while the latter was connected to the counter and counter sense leads of the GAMRY potentiostat. The connections were achieved by using gold (red line) or stainless steel (blue line) wires. The electrodes were introduced in a 100  $\mu\text{M}$  or a 1000  $\mu\text{M}$   $\text{K}_3[\text{Fe}(\text{CN})_6]$  solution and CV was carried out at three different scan rates (5, 50 or 100  $\text{mVs}^{-1}$ ). Figure B.2 shows the second curve produced and it can be observed that the gold wire connection captured the  $\text{K}_3[\text{Fe}(\text{CN})_6]$  curve, while the stainless steel did not under any of the conditions tested. As a result, it was determined that gold wire was the material to be used to connect the working and counter electrodes.



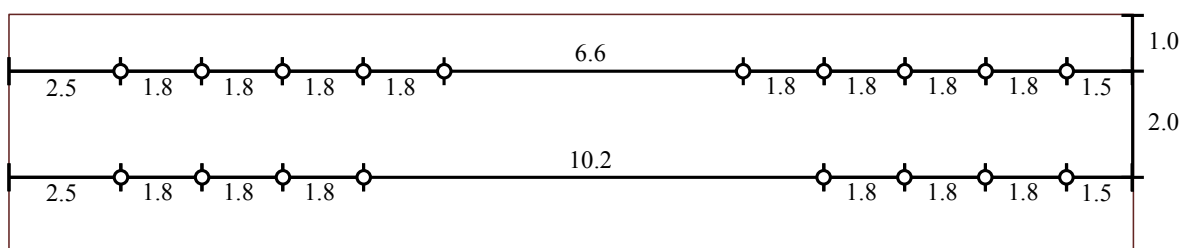
**Figure B.2** Determination of wire material to connect the working and counter electrodes. Second cyclic voltammetry curve obtained by having gold wire (wAu; blue lines) or stainless steel wire (wSS; green lines) to connect as electrode connecting material used to capture the redox reaction of  $K_3[Fe(CN)_6]$ . It can be observed that the gold wire connection was able to capture the reactions, while the stainless steel wire was not.

## B.4 Anaerobic conditions

### SIDE A



### SIDE B



All measurements in cm

1 cm

**Figure B.3** Scaled mask to make the necessary holes on the container walls. File available for true measurements.

## Appendix C

# Anaerobic methods and solutions

### C.1 CCM– related stock solutions

#### C.1.1 Trace metal stock solution (100 X)

In 850 mL of dH<sub>2</sub>O, dissolve 1.5 g nitrilotriacetic acid and adjust the pH to 6.5 with KOH. Then add compounds in Table C.1. Bring final volume to 1L with d H<sub>2</sub>O. Adjust final pH to 7 with HCl and NaOH.

**Table C.1** Trace metal stock solution (100 X, 1 L)

Compound	Amount
MgCl <sub>2</sub> x 6 H <sub>2</sub> O	2.48 g
MnCl <sub>2</sub> x 4 H <sub>2</sub> O	0.585 g
NaCl	1g
FeCl <sub>2</sub> x 4 H <sub>2</sub> O	0.072 g
CoCl <sub>2</sub> x 6 H <sub>2</sub> O	0.152 g
CaCl <sub>2</sub> x 2 H <sub>2</sub> O	0.1 g
ZnCl <sub>2</sub> x 4 H <sub>2</sub> O	0.085 g
CuCl <sub>2</sub>	0.005 g
AlCl <sub>3</sub>	0.01 g
H <sub>3</sub> BO <sub>3</sub>	0.01 g
Na <sub>2</sub> MoO <sub>4</sub> x 2 H <sub>2</sub> O	0.01 g
NiCl <sub>2</sub> x 6 H <sub>2</sub> O	0.03 g
Na <sub>2</sub> SeO <sub>3</sub> x 5 H <sub>2</sub> O	0.0003 g
Na <sub>2</sub> WO <sub>4</sub> x 2 H <sub>2</sub> O	0.008 g

#### C.1.2 Vitamin stock solution (1000 X)

In 1L of dH<sub>2</sub>O dissolve compounds in Table C.2. Vitamins were filter sterilised into a sterile anaerobic serum flask (30 mL in 50 mL flask), crimp sealed and degassed by flushing the headspace of the vial for 30 minutes with oxygen free nitrogen at a flow rate of 0.5 L min<sup>-1</sup> through blue cannulas (0.6 mm ID, Microlance, Beckton Dickinson, Franklin Lakes, NJ, USA) equipped with a sterile filter (Minisart, Sartorius, Göttingen, Germany) on the gassing line.

#### C.1.3 Cysteine–HCl stock solution (100 X)

The 100 X cysteine stock solution had a concentration of 35.03 g / L stock to achieve a final concentration of 0.35 g / L. The solution was prepared by filling the volume of ultrapure water

**Table C.2** Vitamin stock solution (1000 X, 1 L)

Compound	Amount
Biotin	2 mg
Folic acid	2 mg
Pyridoxin-HCl	10 mg
Thiamine-HCl x 2H <sub>2</sub> O	5 mg
Riboflavin	5mg
Nicotinic acid	5 mg
Vitamin B12	0.1 mg
p-Aminobenzoic acid	5 mg
Lipoic acid	5 mg

and the corresponding amount of Cysteine-HCl x H<sub>2</sub>O into two different tubes. These were both transferred into the anaerobic chamber overnight to degas. The water and salt were combined in one tube and mixed by inverting until the salt had completely dissolved. The solution was transferred into a serum flask, which was then seal with a rubber stopper, capped and autoclaved.

### C.1.4 Na<sub>2</sub>S stock solution (50 X)

The 50 X sodium sulphide (Na<sub>2</sub>S) stock solution had a concentration of 24 g / L stock (0.1 M), for a final concentration of 0.48 g / L (2 mM). The required amount of Na<sub>2</sub>S x 9 H<sub>2</sub>O was weighed carefully. The crystals were washed with ultrapure H<sub>2</sub>O and dried on a piece of microscope paper (to avoid fibres) on top of a piece of kitchen roll. The washed Na<sub>2</sub>S x 9 H<sub>2</sub>O were weighed again and placed in a tube and a second tube was filled with the corresponding amount of water. Both tubes were transferred into the anaerobe chamber and left overnight to degas. The H<sub>2</sub>O was poured into the tube containing the Na<sub>2</sub>S. The solution was mixed well and transferred into a serum vial. After sealing it with a stopper and capping it, it was autoclaved.

## C.2 Protocol for CCM preparation

The medium was prepared in batches of 0.5 L by mixing the basal salt solution and adding the trace metal stock solution (100 X) and resazurin stock solution (1 g / L, 1000 X) in a 1 L bottle (Duran, DE). Sodium lactate was added for a final 30 mM concentration. The pH was adjusted to 7 with 1 M HCl. The medium was heated in a microwave (800W, 30 s / 100 mL). The bottle was then transferred to a water bath set to ca. 80 °C. Continuous flow of 80/20% N<sub>2</sub>/CO<sub>2</sub> (anoxic) was maintained at 0.5 L min<sup>-1</sup> flow rate into the liquid phase using tubing and a rubber stopper to close off the top opening. After 30 min, the vitamin (1000 X) and cysteine-HCl (100 X) stock solutions were added. The removal of oxygen was verified by a colour-shift from pink to colourless by the resazurin (usually after 90 min of degassing). The gas tubing was quickly removed while continuing the gas flow and the rubber stopper was quickly fixed on bottle. The reduction potential achieved was below -300 mV. A pressure cap was placed on the bottle mouth to ensured that the rubber bun would stay in place. The medium contained in the bottle was then autoclaved (121 °C, 15 min) using a desktop autoclave (ST 19 T, Dixon, Wickford, UK). If the CCM had to be aliquoted, the CCM was first moved into the anaerobic chamber (MACS-MG-500 anaerobic workstation, Don Whitley Scientific, Shipley, UK). Hungate tubes (27 mL Balch tubes, Chemglass Life Sciences, Vineland, NJ, USA) and glass serum vials (50 mL and 125 mL capacity), previously degassed for min. 24 h in the anaerobic chamber, were filled with 4.5 mL and 25 mL, respectively. Both the tubes and the serum vials were closed with blue butyl rubber septa (Chemglass Life

Sciences, Vineland, NJ, USA), crimp sealed and autoclaved.

### C.3 Headspace replacement method

The headspace of Hungate tubes or serum vials was replaced by inserting a 0.2  $\mu\text{m}$  filter to a needle and placing the filter at the end of a manifold line. The desired gas was opened in the manifold with the pressure set to 2 bar and set to flow to ca. 0.5 L  $\text{min}^{-1}$ . With the gas flowing, the needle was inserted into the rubber cap. Once the flow in the manifold dropped to zero, indicating that the Hungate tube or vial pressure had equilibrated with the manifold pressure, a second needle was introduced into the rubber cap to allow gas to flow out. After 3 min, the second needle was removed. Once the flow dropped to zero again, the manifold line was removed from the rubber cap and then turned off. All gases used for headspace flushing were run through an oxygen scrubber column (Oxisorb, MG Industries, Bad Soden, Germany) in the manifold, to remove any residual oxygen.

### C.4 Anaerobic cryostocks preparation protocol

The protocol presented here is a modification of the protocol used in the Molecular Cell Physiology Department at the Vrije Universiteit titled “*Glycerol stocks of anaerobic bacteria for storage at -80 °C*” ([https://www.bio.vu.nl/~microb/Protocols/Media\\_and\\_solutions/glycerol\\_stocks\\_anaerobic.pdf](https://www.bio.vu.nl/~microb/Protocols/Media_and_solutions/glycerol_stocks_anaerobic.pdf) last accessed 22/08/2018).

#### C.4.1 Solution preparation

##### C.4.1.1 0.1 M phosphate buffer pH 7 (200 mL)

Prepare **sol A** (3.12 g  $\text{NaH}_2\text{PO}_4$  in 100 mL ultrapure water) and **sol B** (3.56 g  $\text{Na}_2\text{HPO}_4$  in 100 mL ultrapure water). Mix 39 mL sol A with 61 mL sol B and add 100 mL ultrapure water.

##### C.4.2 ~100 mM titanium citrate solution (325 mL)

*Method from Zehnder and Wuhrmann (1976).* Prepare 250 mL of 0.2 M  $\text{Na}_3$ -citrate (14.705 g / 250 mL) and heat in microwave (30 s / 100 mL). Degas with  $\text{N}_2$  at 0.5 L  $\text{min}^{-1}$  for 1 h. In the anaerobic chamber, add 25 mL 15%  $\text{TiCl}_3$  and 50 mL 8%  $\text{Na}_2\text{CO}_3$  (w/v) (8 g / 100mL). Mix and filter-sterilize.

##### C.4.2.1 50% Glycerol solution (v/v)

Measure 100 mL of the 0.02 M Phosphate buffer (pH 7). Add 0.1 mL resazurin (final concentration 0.5 mg/l). Heat the solution in microwave (30 s / 100 mL) and cool down under  $\text{N}_2$  stream. Add 100 ml glycerol (50% (v/v) glycerol solution). Fill serum bottles with the glycerol solution. Stopper and cap them. Change headspace to 100%  $\text{N}_2$  or 80/20%  $\text{N}_2/\text{CO}_2$  and fill with 0.2 atm overpressure. Autoclave the serum bottles.

#### C.4.3 Procedure

Mix a late log phase bacterial culture with the sterilized glycerol solution in a 1:1 ratio. The final concentration of glycerol should be 25 – 30%. Add 3 drops of the Ti-citrate solution using a syringe needle to reduce stocks. Aliquot 1.5 mL into cryovials (2 mL cryo tube with external thread and jacket, 65–9001, FluidX, UK). Freeze at -80 °C.

# Appendix D

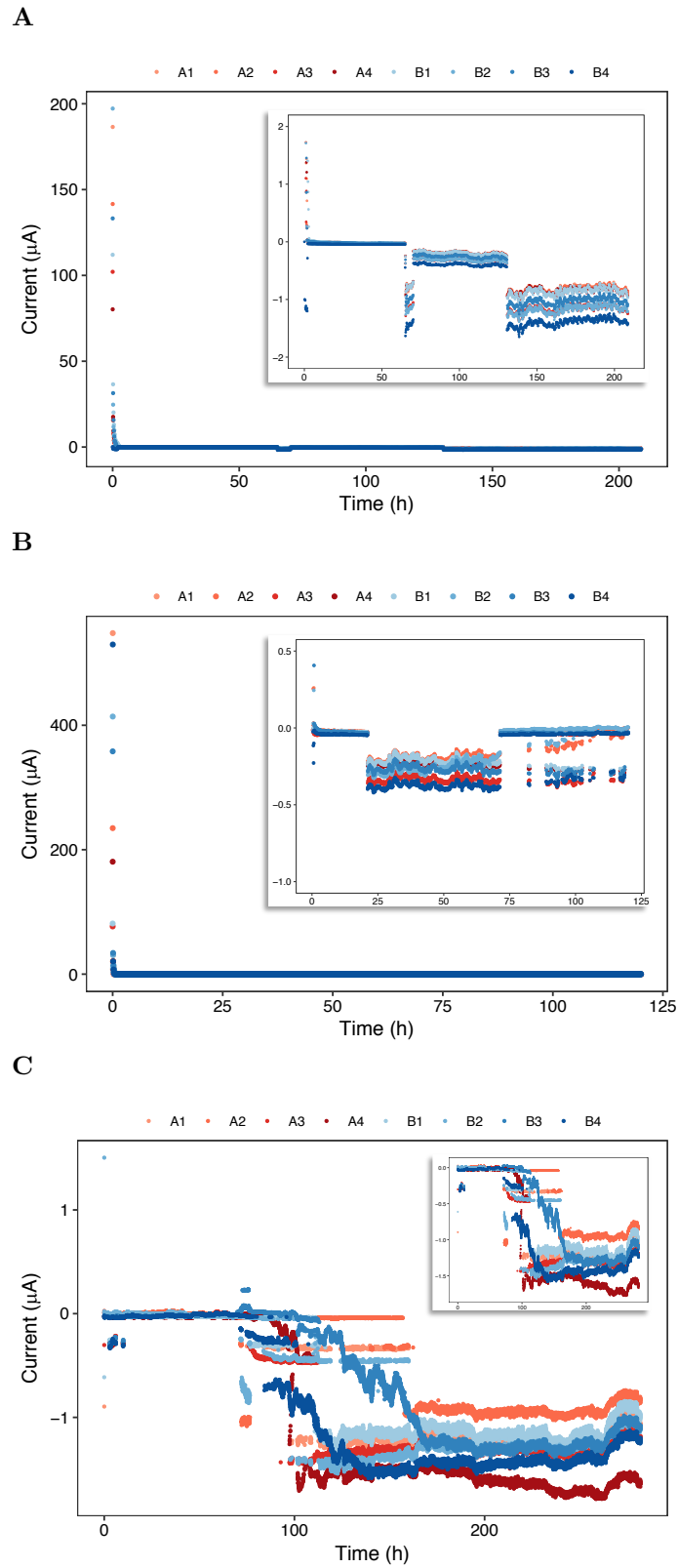
## Electrochemical experiment

### D.1 Current measurements

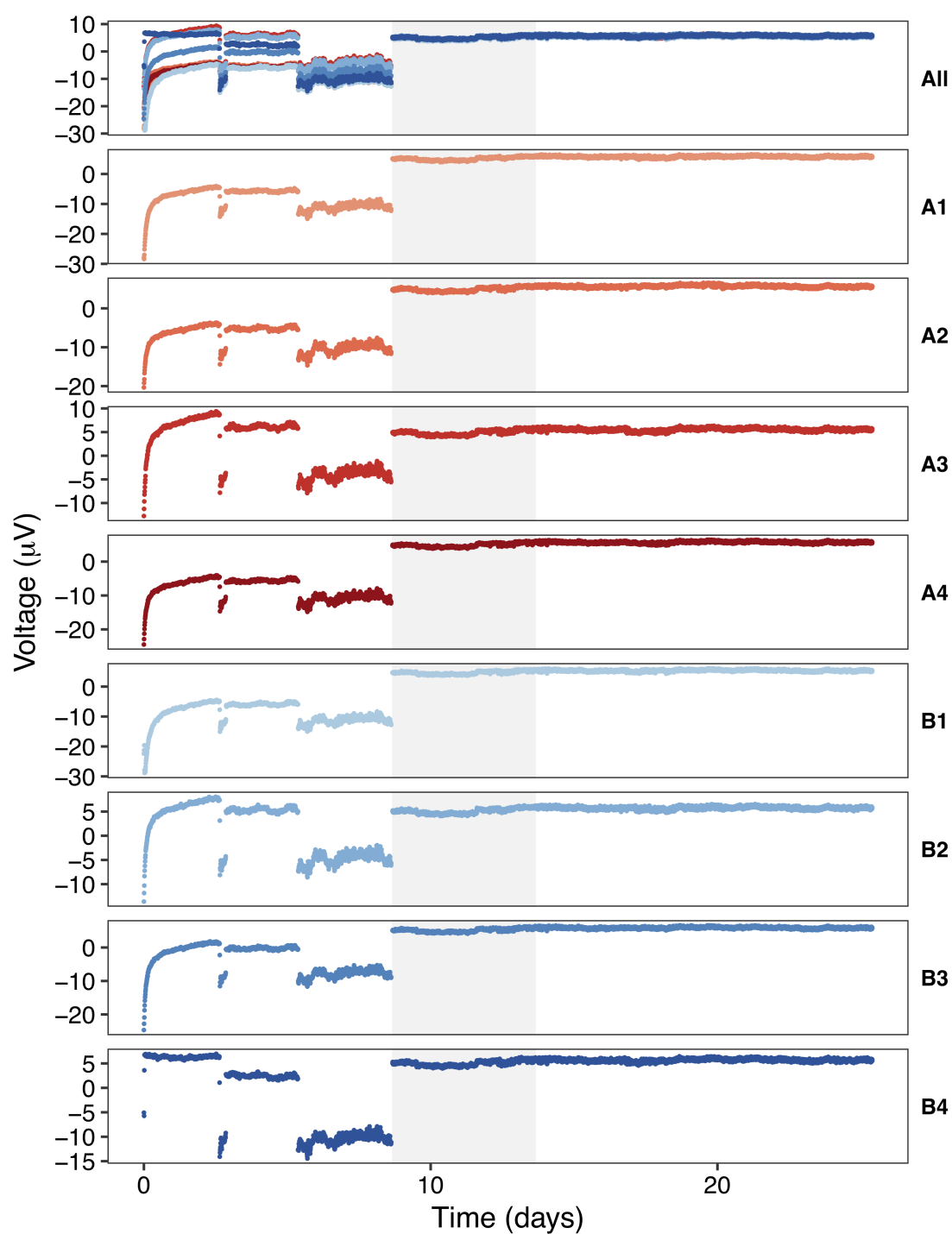
#### D.1.1 Estimation of the number of electrons exchanged

**Table D.1** Number of Coulombs estimated with Equations 3.3, 3.4 and 3.5 for  $nC_{\text{net}}$ ,  $nC_{\text{abs}}$  and  $nC_{\text{diff}}$ , respectively.

Electrochemical cell	$nC$	$nC_{\text{abs}}$	$nC_{\text{diff}}$
<b>A1</b>	-0.879230	1.250552	0.259547
<b>A2</b>	-0.667475	0.891083	0.213619
<b>A3</b>	-1.302093	1.434762	0.132670
<b>A4</b>	-1.281311	1.457601	0.176290
<b>B1</b>	-0.962734	1.199639	0.236905
<b>B2</b>	-1.097681	1.441439	0.283117
<b>B3</b>	-0.911332	1.235530	0.301902
<b>B4</b>	-1.462275	1.640467	0.176159



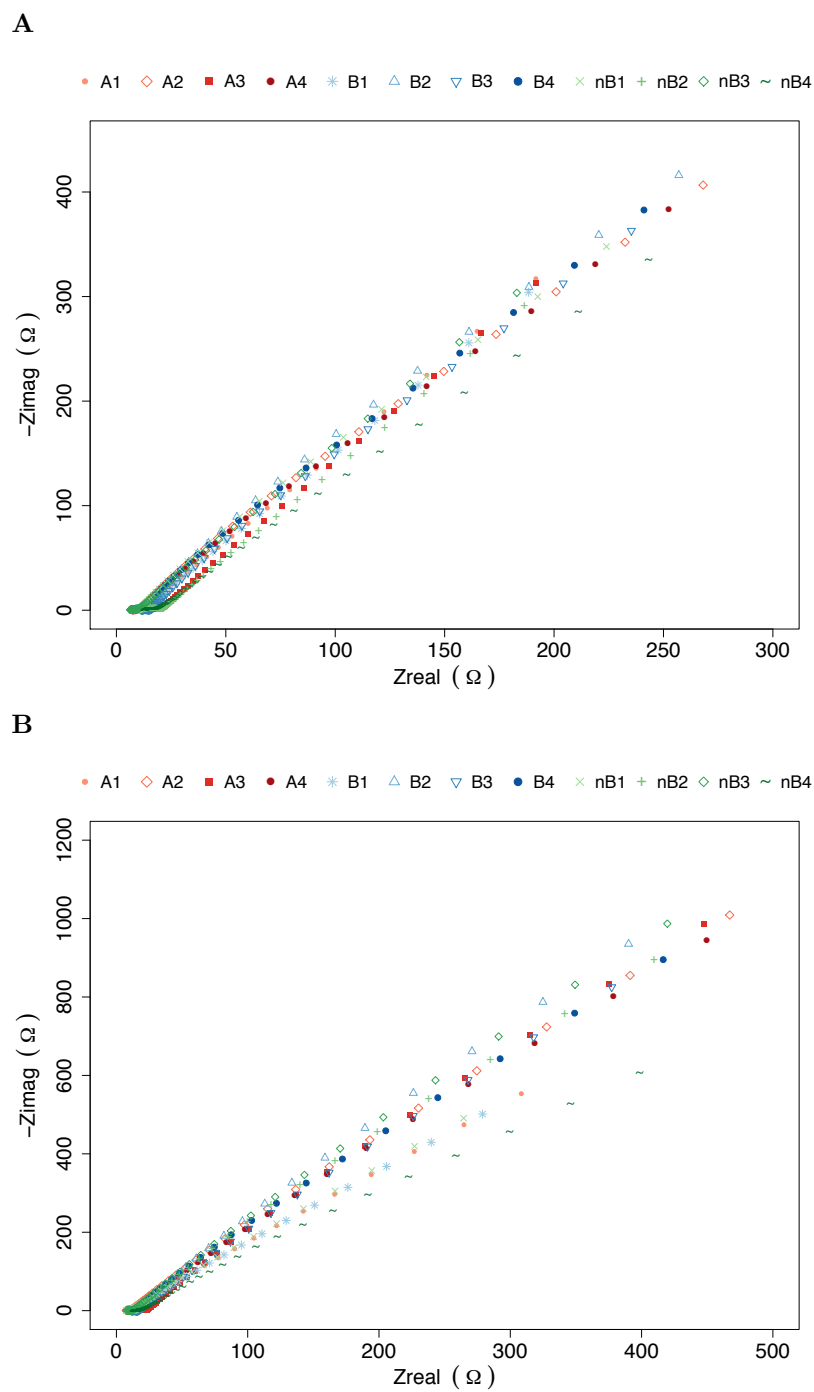
**Figure D.1** Current over time for the three periods monitored. A, abiotic (reds); B, biotic (Dv and Mm inoculated and connected; blues); nB, biotic not connected (Dv and Mm inoculated, but not connected; greens).



**Figure D.2** Voltage across W1 and W2 recorded over time during the current measurement shown for the different electrochemical cells. Note the difference in y-axis ranges. As the potential is dynamically set to 0 V, a voltage close to 0 V is expected to be measured, with 30  $\mu\text{V}$  being the minimum potential that can be measured reliably (priv. comm. with potentiostat manufacturer). SHE, standard hydrogen electrode; A, abiotic (reds); B, biotic (Dv and Mm inoculated and connected; blues); nB, biotic not connected (Dv and Mm inoculated, but not connected; greens). The number refers to the replicate number.

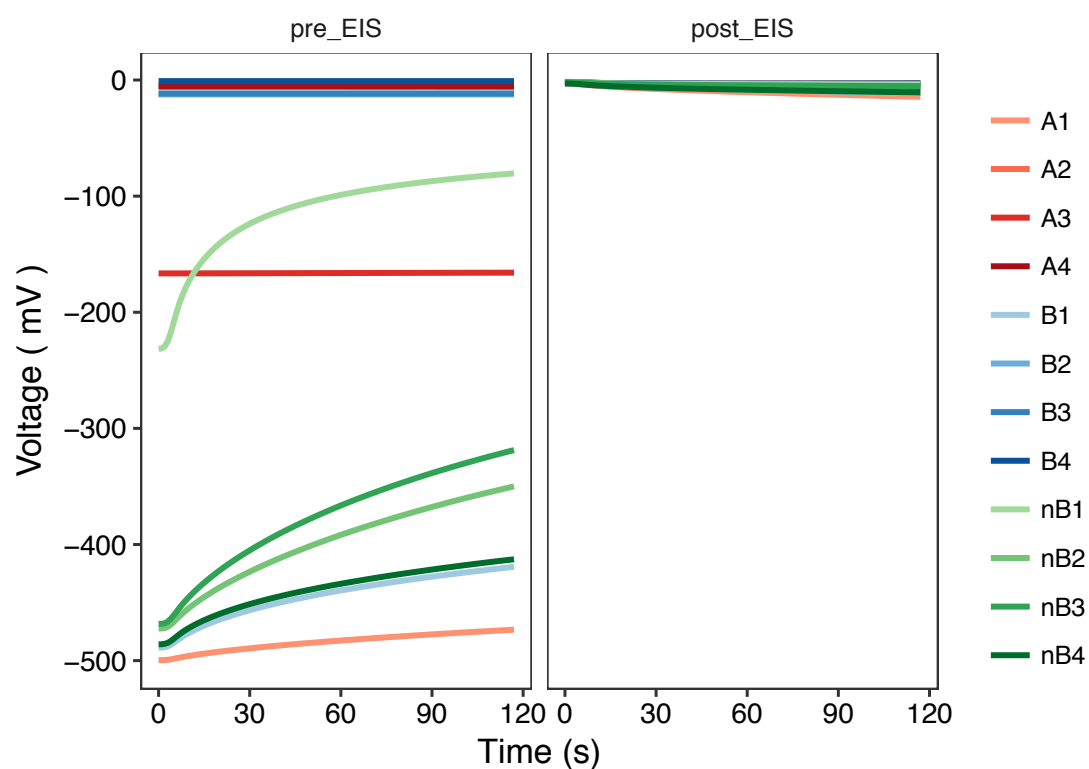


## D.2 EIS

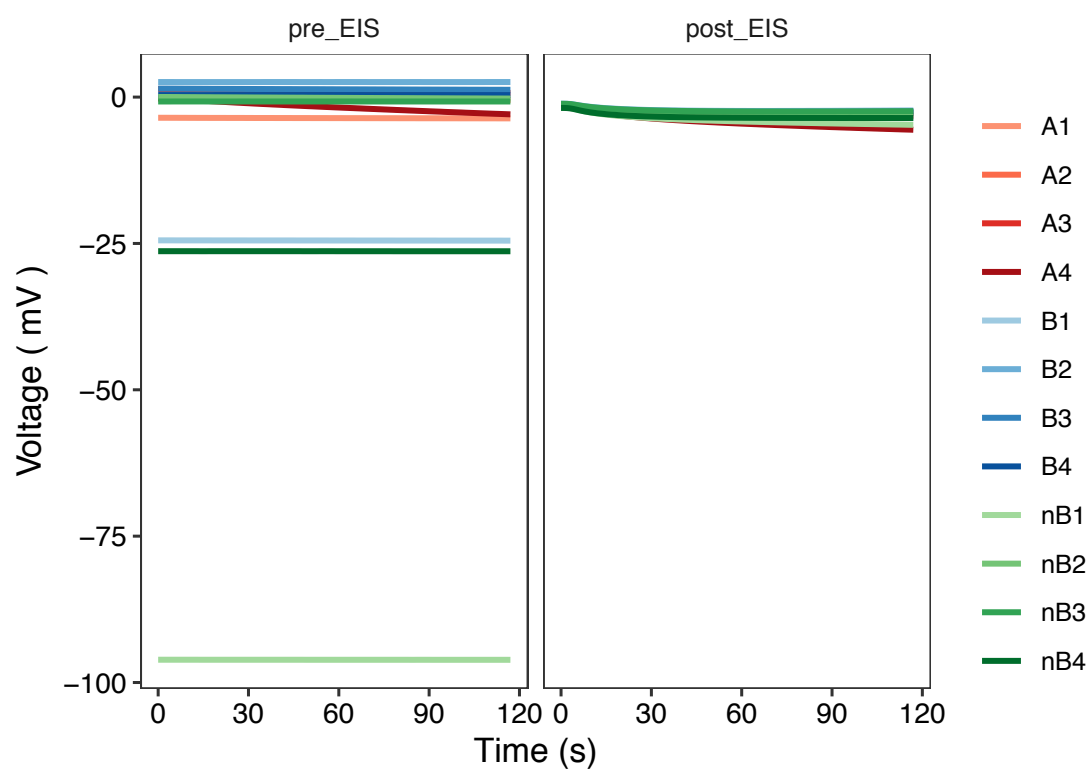


**Figure D.3** EIS nyquist. Full range of data values obtained. A, abiotic (reds); B, biotic (Dv and Mm inoculated and connected; blues); nB, biotic not connected (Dv and Mm inoculated, but not connected; greens).

A

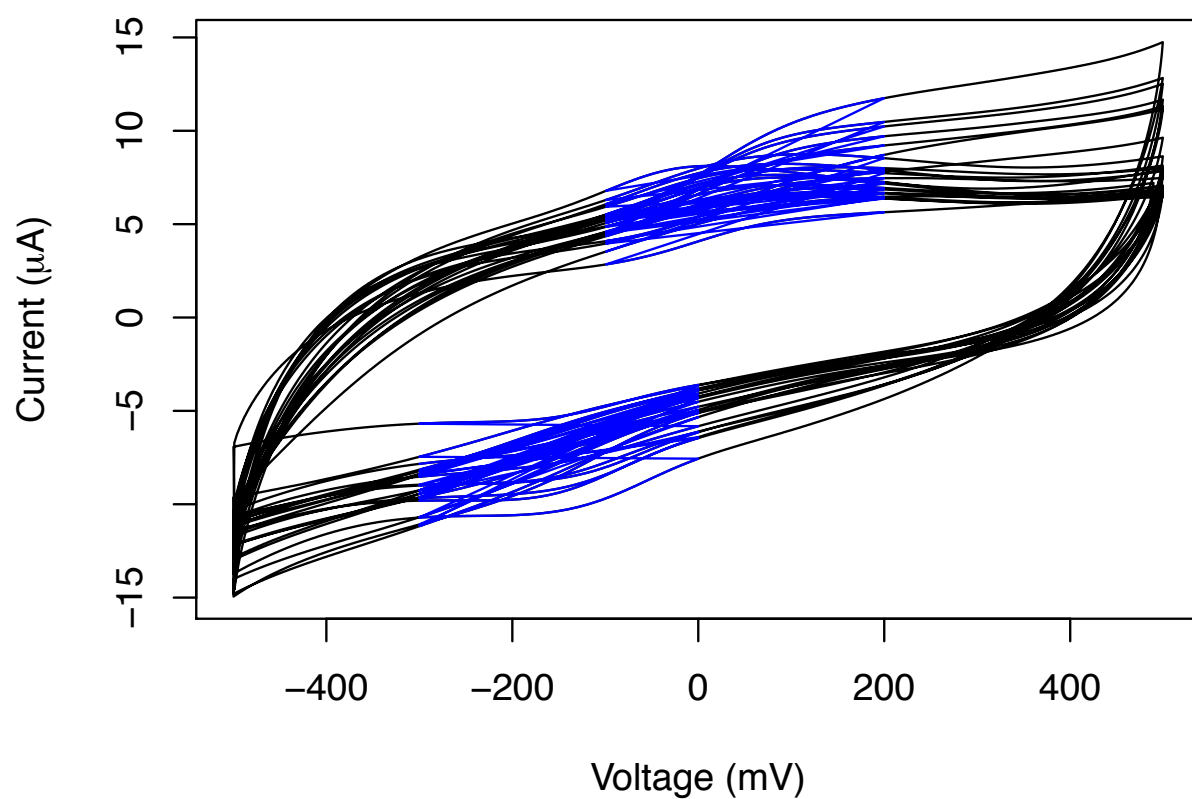


B

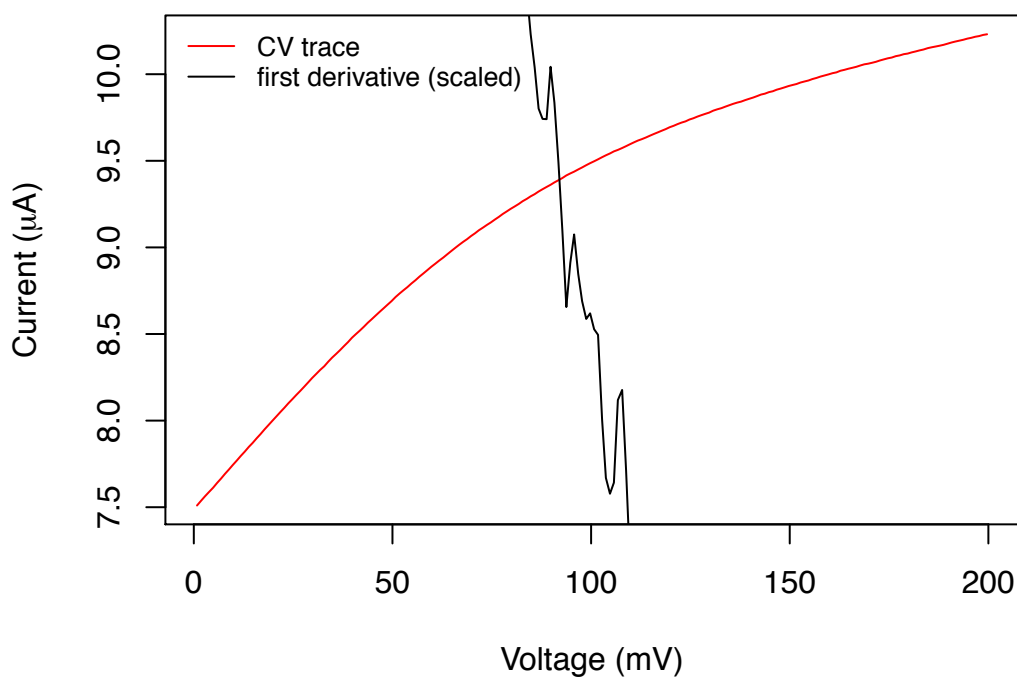


**Figure D.4** OCP before and after EIS. A, abiotic (reds); B, biotic (Dv and Mm inoculated and connected; blues); nB, biotic not connected (Dv and Mm inoculated, but not connected; greens).

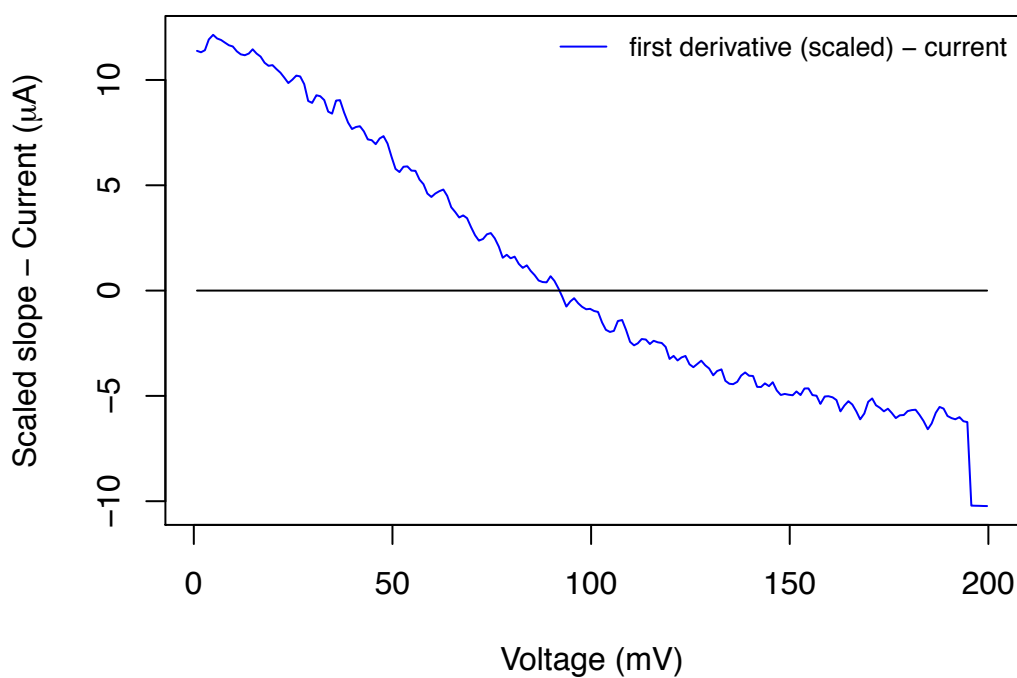
### D.3 CV



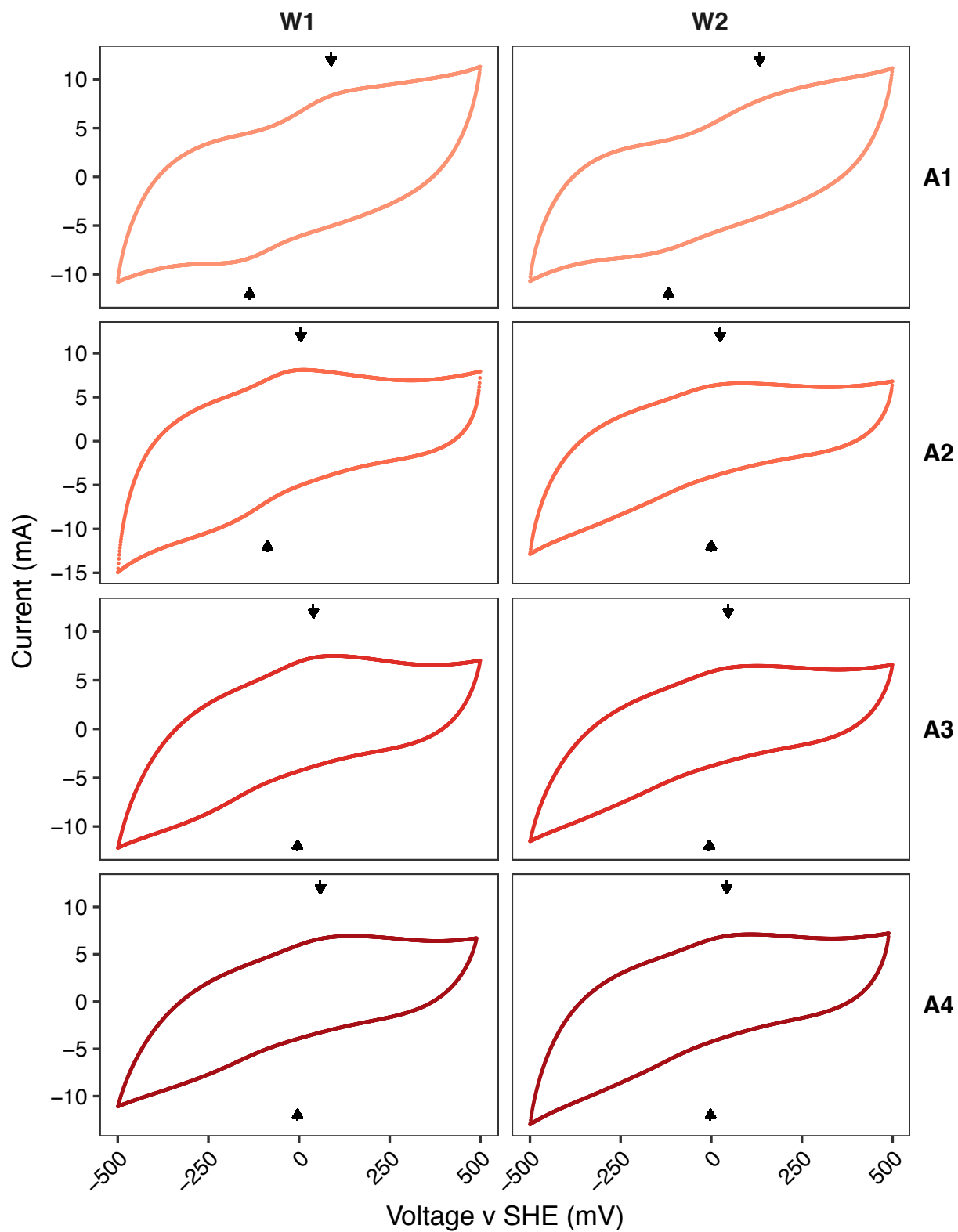
**Figure D.5** Range of cyclic voltammetry trace (fifth curve) used to find the peak potentials. All the fifth curve of all the CV traces performed in this work were overlaid. The blue sections highlight the regions that were used to find the potential peaks. The regions to identify the anodic ( $E_{p,a}$ ) and cathodic ( $E_{p,c}$ ) potential peak were in the negative and positive current range, respectively. Refer to Section 3.4.10. Voltage reported v SHE (standard hydrogen electrode).



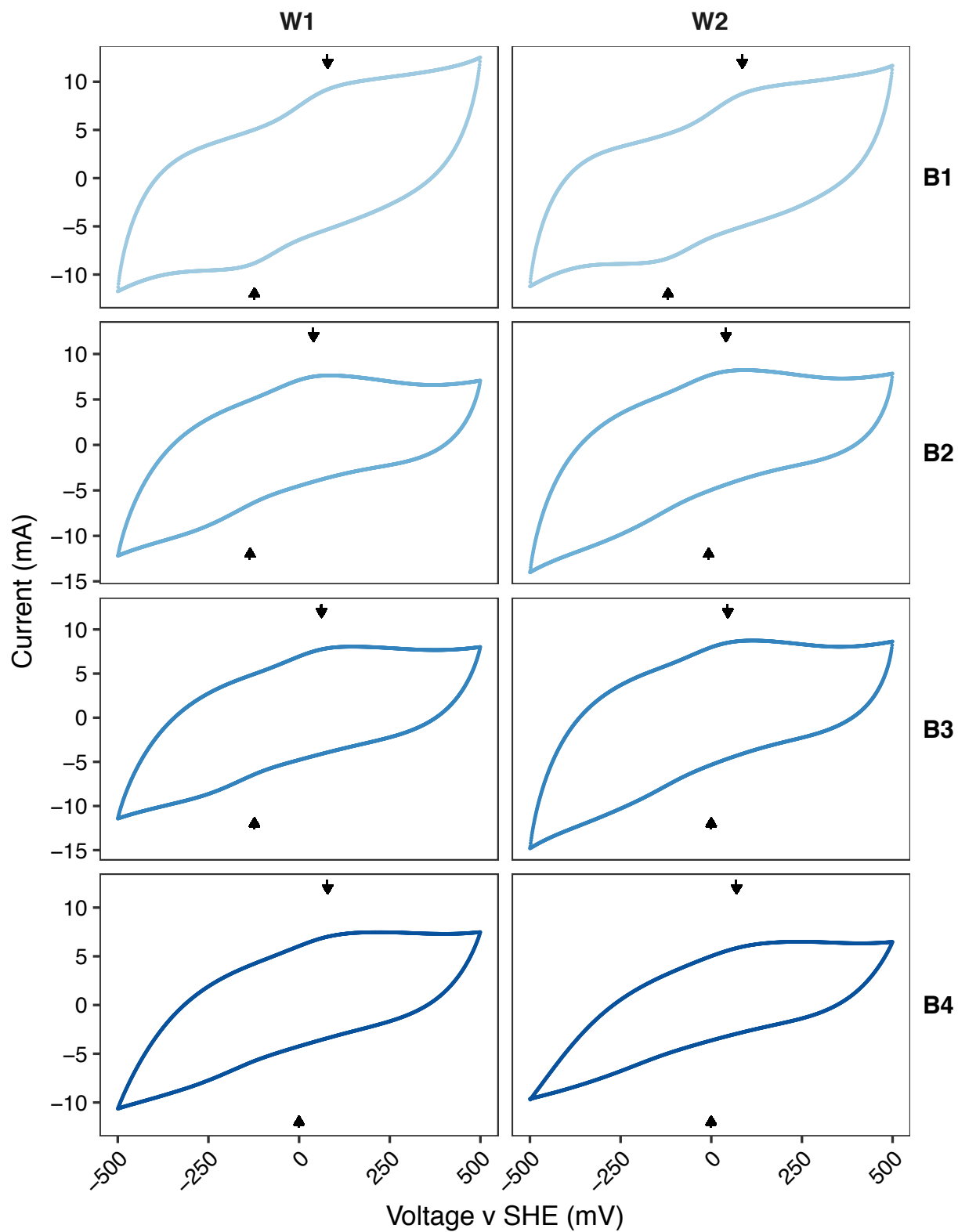
**Figure D.6** CV peak determination: CV trace and first derivative. Refer to Section 3.4.10. Voltage reported v SHE (standard hydrogen electrode).



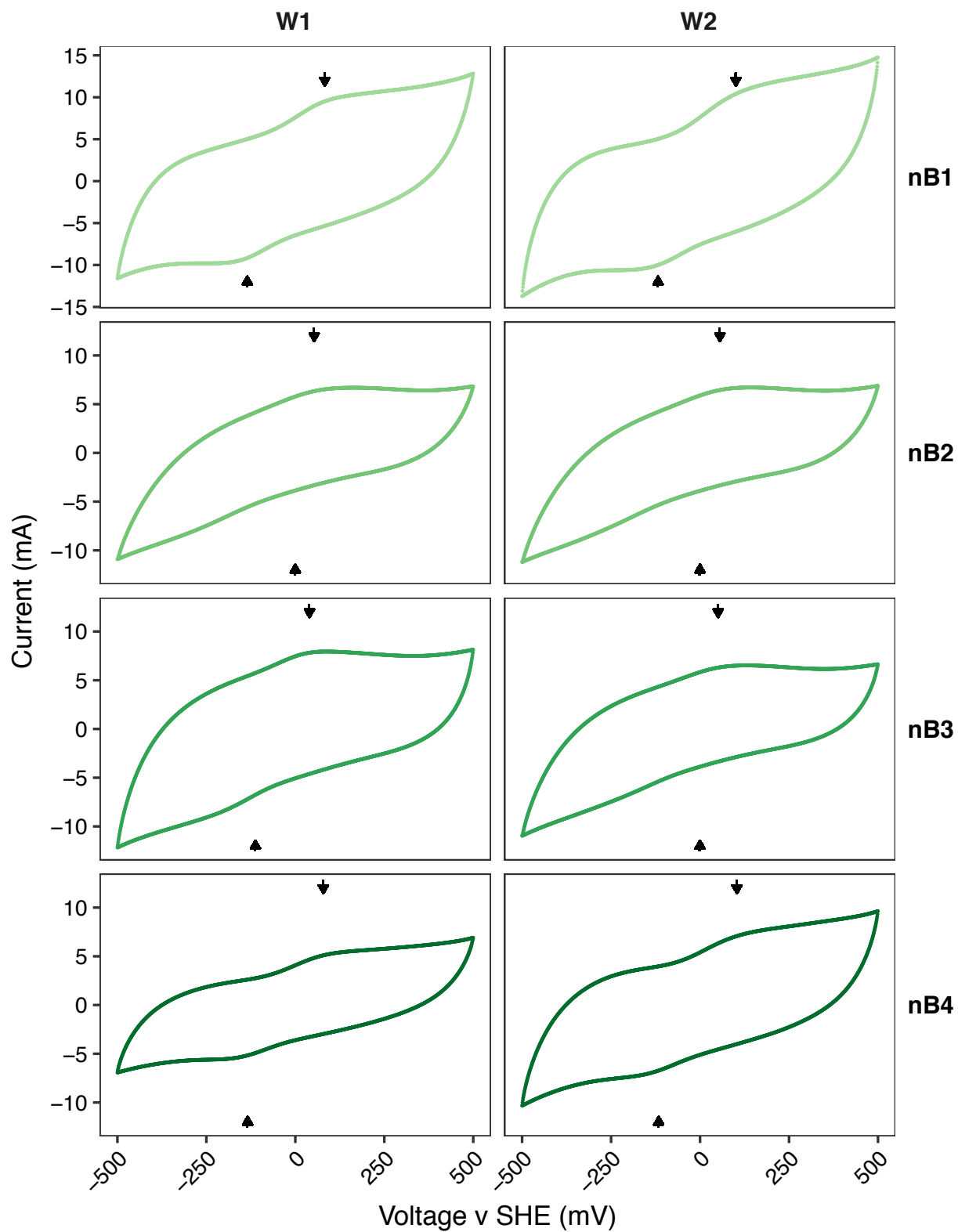
**Figure D.7** CV peak determination: scaled slope minus the current in range. The y-axis refers to the difference between the current and the scaled slope as discussed in Section 3.4.10. The black line is placed at " $y = 0$ " to visually identify the location of the lines shown on Figure D.6 corresponding to where the blue line crosses the black. Note that  $y$  refers to the y-axis, not a variable. Refer to Section 3.4.10. Voltage reported v SHE (standard hydrogen electrode).



**Figure D.8** Cyclic voltammetry (CV) trace of the fifth curve for each working electrode on the abiotic (A) electrochemical cells. The arrows indicate the anodic ( $\uparrow$ ,  $E_{p,a}$ ) and cathodic ( $\downarrow$ ,  $E_{p,c}$ ) peak potentials estimated with the `first_derivative_inflection()` function (Code 3.1).



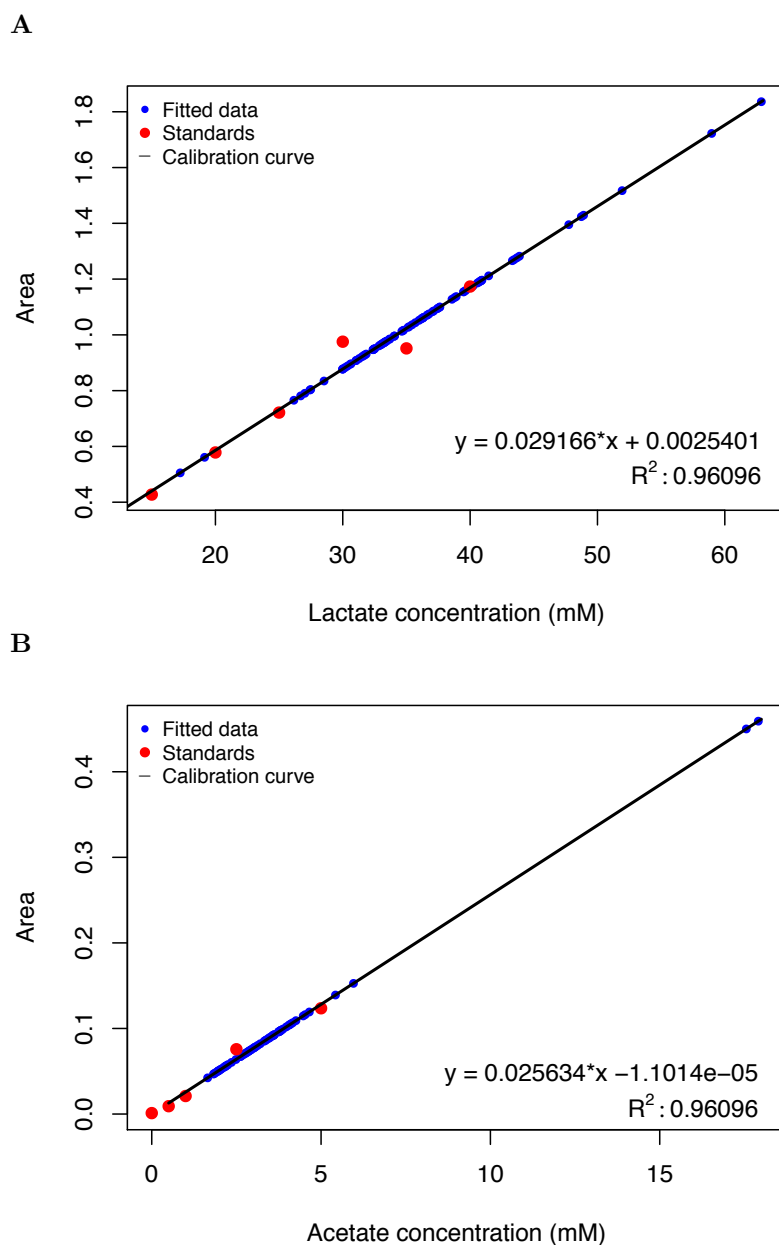
**Figure D.9** Cyclic voltammetry (CV) trace of the fifth curve for each working electrode on the biotic (B) electrochemical cells. The arrows indicate the anodic ( $\uparrow$ ,  $E_{p,a}$ ) and cathodic ( $\downarrow$ ,  $E_{p,c}$ ) peak potentials estimated with the `first_derivative_inflection()` function (Code 3.1).



**Figure D.10** Cyclic voltammetry (CV) trace of the fifth curve for each working electrode on the non-connected biotic (nB) electrochemical cells. The arrows indicate the anodic ( $\uparrow$ ,  $E_{p,a}$ ) and cathodic ( $\downarrow$ ,  $E_{p,c}$ ) peak potentials estimated with the `first_derivative_inflection()` function (Code 3.1).

## D.4 Ion chromatography

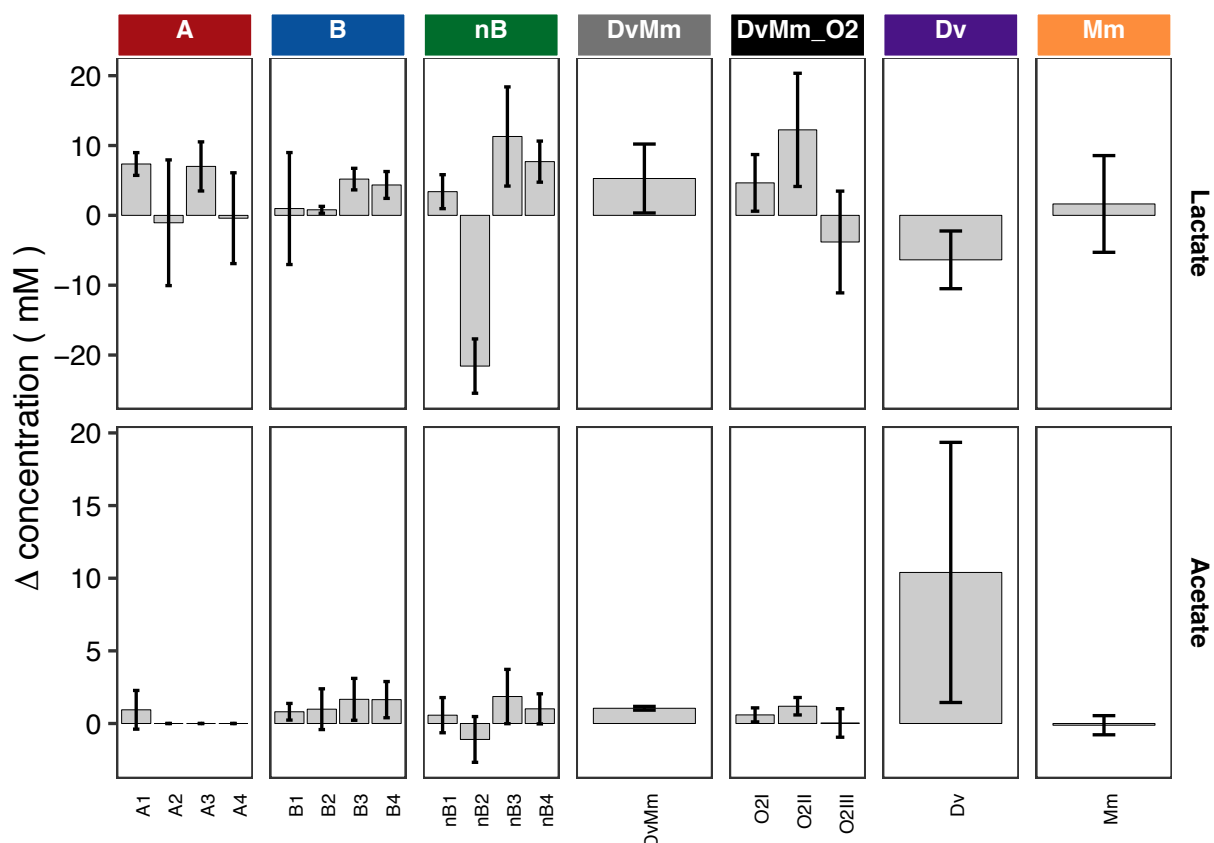
### D.4.1 Standard curve and calculation of the sample concentrations



**Figure D.11** Standard curve and calculation of the sample concentrations for lactate (**A**) and acetate (**B**). The standards indicated with red dots were used to do a linear regression (black line) to find the relationship between area measured in the chromatogram and the compound concentrations. The line equation and goodness of fit ( $R^2$ ) can be seen in the graphs. The corresponding equation were used to calculate the concentrations of the samples. Note that extrapolation was required in both cases due to unexpected high compound concentrations in the unknown samples. This was likely due to evaporation and in future, the chloride peak could be use as a correction, as discussed in the main text. See Materials and Methods Section 3.4.11.



## D.4.2 Change in compound concentration for the samples



**Figure D.12** Change ( $\Delta$ ) in compound concentration for the different samples. A, abiotic; B, biotic (Dv and Mm inoculated and connected); nB, biotic not connected (Dv and Mm inoculated, but not connected); DvMm, control DvMm cultures in sealed Hungate tubes; DvMm\_O2, control DvMm cultures with permeable membranes introduced into the containers, identified by I, II and III; Dv, control Dv cultures in sealed Hungate tubes; Mm, control Mm cultures in sealed Hungate tubes. Error bars show the standard deviation (sd) with  $n = 2$  for A, B and nB and  $n = 3$  for the control tubes.

## D.4.3 Statistical analyses of the change of sample concentrations

### D.4.3.1 Lactate

#### ONE WAY ANOVA: ALL CONDITIONS

A, B, Dv, DvMm, Mm, nB, DvMm\_O2

ANOVA ANALYSIS – FORMULA: CONC  $\sim$  CONDITION

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
CONDITION	6.000000	335.127934	55.854656	0.749178	0.614168
Residuals	35.000000	2609.409273	74.554551		

#### PAIRED T TEST: HALF-CELL COMPARISON BY CONDITION

A, B, Dv, DvMm, Mm, nB, DvMm\_O2

Paired t-test

data: COND\_A\$CONC[HC\_A\_index] and COND\_A\$CONC[HC\_B\_index]

t = 0.044685, df = 3, p-value = 0.9672

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-15.05920 15.48812

sample estimates:

mean of the differences

0.2144593

Paired t-test

data: COND\_B\$CONC[HC\_A\_index] and COND\_B\$CONC[HC\_B\_index]

t = -0.81368, df = 3, p-value = 0.4754

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-12.408975 7.355621

sample estimates:

mean of the differences

-2.526677

Paired t-test

data: COND\_nB\$CONC[HC\_A\_index] and COND\_nB\$CONC[HC\_B\_index]

t = 0.57228, df = 3, p-value = 0.6072

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-9.054888 13.025428

sample estimates:

mean of the differences

1.98527

#### D.4.3.2 Acetate

##### **ONE WAY ANOVA: ALL CONDITIONS**

A, B, Dv, DvMm, Mm, nB, DvMm\_O2

ANOVA ANALYSIS – FORMULA: CONC ~ CONDITION

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
CONDITION	6.000000	271.972564	45.328761	8.166022	0.000015
Residuals	35.000000	194.281446	5.550898		

POST HOC: Tukey multiple comparisons of means  
95% family-wise confidence level

	term	comparison	estimate	conf.low	conf.high	adj.p.value	symbol
1	CONDITION	B-A	1.039193	-2.643213	4.721599	0.972944	
2	CONDITION	Dv-A	10.162356	5.176355	15.148357	0.000005	***
3	CONDITION	DvMm-A	0.812495	-4.173506	5.798496	0.998551	
4	CONDITION	DvMm_O2-A	0.370297	-3.208358	3.948953	0.999894	
5	CONDITION	Mm-A	-0.353877	-5.339879	4.632124	0.999988	
6	CONDITION	nB-A	0.349758	-3.332648	4.032164	0.999936	
7	CONDITION	Dv-B	9.123164	4.137162	14.109165	0.000035	***
8	CONDITION	DvMm-B	-0.226697	-5.212699	4.759304	0.999999	
9	CONDITION	DvMm_O2-B	-0.668895	-4.247551	2.909760	0.996892	
10	CONDITION	Mm-B	-1.393070	-6.379071	3.592931	0.974248	
11	CONDITION	nB-B	-0.689435	-4.371841	2.992971	0.996863	
12	CONDITION	DvMm-Dv	-9.349861	-15.363205	-3.336517	0.000451	***
13	CONDITION	DvMm_O2-Dv	-9.792059	-14.701933	-4.882184	0.000007	***
14	CONDITION	Mm-Dv	-10.516234	-16.529577	-4.502890	0.000075	***
15	CONDITION	nB-Dv	-9.812598	-14.798600	-4.826597	0.000010	***
16	CONDITION	DvMm_O2-DvMm	-0.442198	-5.352072	4.467677	0.999953	
17	CONDITION	Mm-DvMm	-1.166373	-7.179716	4.846971	0.996191	
18	CONDITION	nB-DvMm	-0.462737	-5.448738	4.523264	0.999944	
19	CONDITION	Mm-DvMm_O2	-0.724175	-5.634049	4.185700	0.999176	
20	CONDITION	nB-DvMm_O2	-0.020539	-3.599195	3.558116	1.000000	
21	CONDITION	nB-Mm	0.703635	-4.282366	5.689637	0.999359	

## ***PAIRED T TEST: HALF-CELL COMPARISON BY CONDITION***

A, B, Dv, DvMm, Mm, nB, DvMm\_O2

Paired t-test

data: COND\_A\$CONC[HC\_A.index] and COND\_A\$CONC[HC\_B.index]

t = 1, df = 3, p-value = 0.391

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-1.028066 1.970189

sample estimates:

mean of the differences

0.4710614

Paired t-test

data: COND\_B\$CONC[HC\_A.index] and COND\_B\$CONC[HC\_B.index]

t = 5.7721, df = 3, p-value = 0.01034

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

0.7399204 2.5585297

sample estimates:

mean of the differences

1.649225

Paired t-test

data: COND\_nB\$CONC[HC\_A.index] and COND\_nB\$CONC[HC\_B.index]

t = 7.6276, df = 3, p-value = 0.004678

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

1.172258 2.850773

sample estimates:

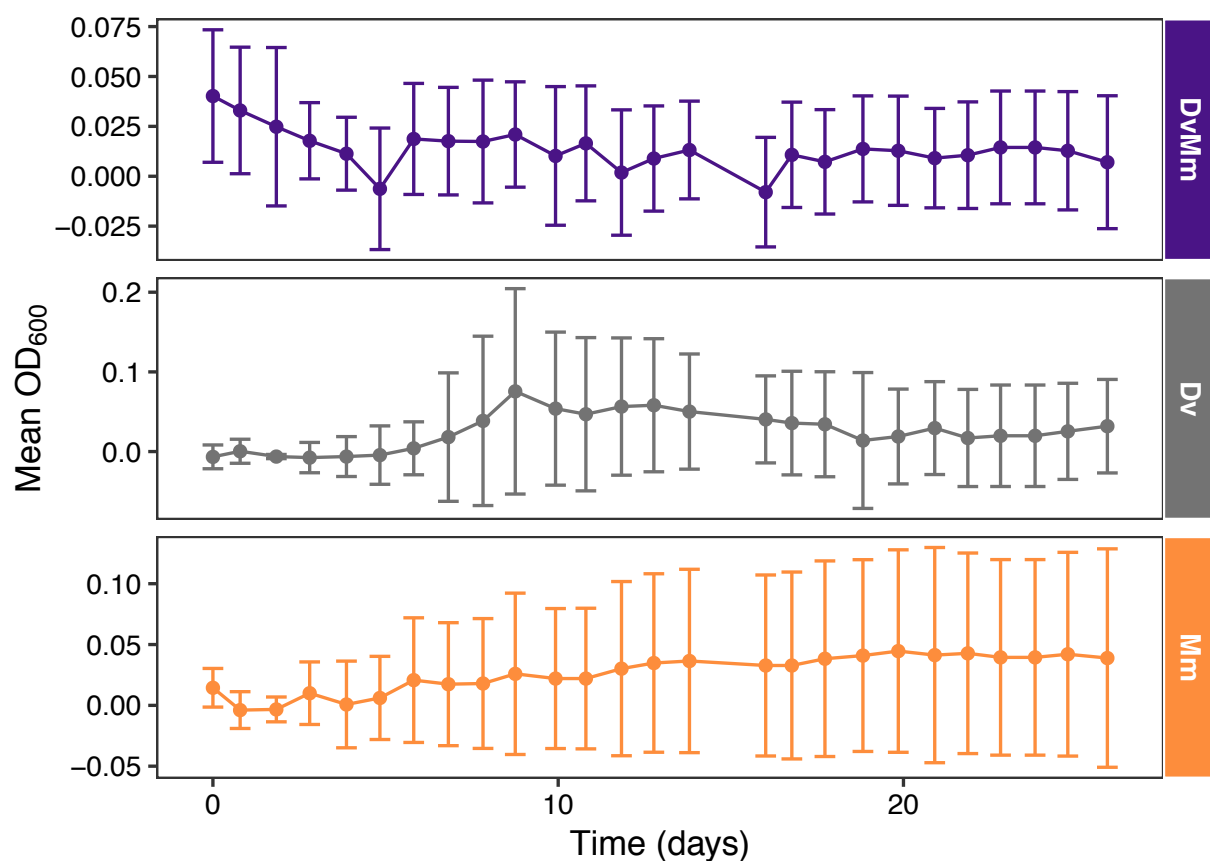
mean of the differences

2.011516

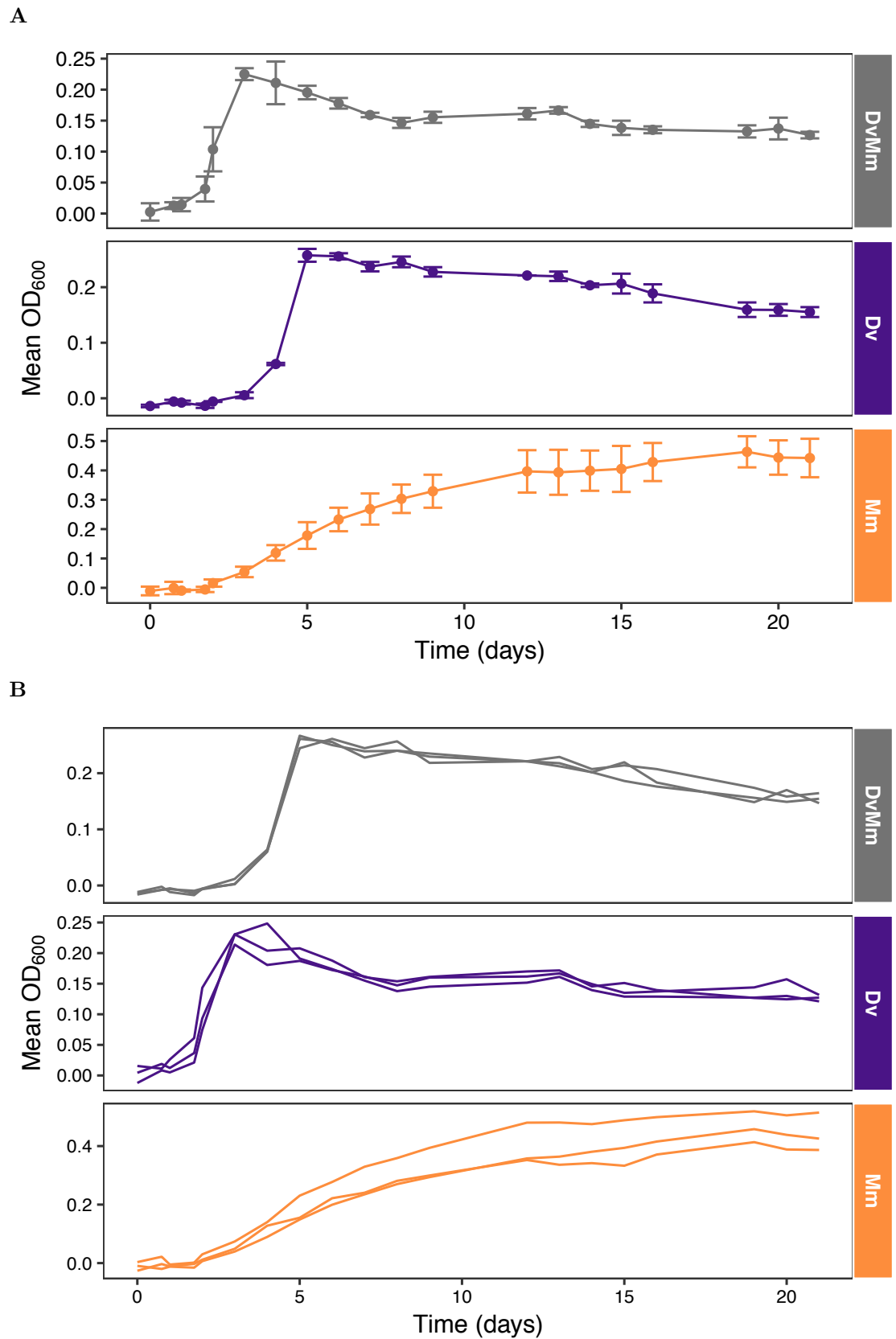
### **D.4.4 Growth analyses of the control cultures**

Figure D.13 shows the mean OD<sub>600</sub> with the standard deviation as error bars for the control cultures grown in Hungate tubes and monitored throughout the experiment. Growth curves of the microorganisms grown at 37 °C using the same strains and medium as reported in this work are shown in Figure D.14. This is shown as both the mean OD<sub>600</sub> with the standard deviation

as error bars (top) and the individual replicate curves (bottom). The data was generated by Dr. Jing Chen (unpublished).



**Figure D.13** Growth curves of control cultures at room temperature (ca. 21 °C). The graphs show the mean OD<sub>600</sub> over time. Error bars show the standard deviation (sd) with  $n = 3$ . The low OD<sub>600</sub> achieved means that a high standard deviation was measured and hence the graph is not a clear representation of the growth achieved. See Figure 3.11 for the trace of each replicate.



**Figure D.14** Growth curves of culture tubes at 37 °C showing the mean OD<sub>600</sub> with the standard deviation as the error bars **A** or the individual cultures **B**. Data generated by Dr. Jing Chen (unpublished).

## Appendix E

### **MetQy**



### Data and text mining

## MetQy—an R package to query metabolic functions of genes and genomes

Andrea S. Martinez-Vernon<sup>1,2,3</sup>, Frederick Farrell<sup>2</sup> and  
Orkun S. Soyer<sup>1,2,3,\*</sup>

<sup>1</sup>Synthetic Biology Centre for Doctoral Training and <sup>2</sup>School of Life Sciences and <sup>3</sup>Warwick Integrative Synthetic Biology (WISB) Centre, Life Sciences Building, University of Warwick, Coventry CV4 7AL, UK

\*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on November 26, 2017; revised on March 14, 2018; editorial decision on May 25, 2018; accepted on June 1, 2018

### Abstract

**Summary:** With the rapid accumulation of sequencing data from genomic and metagenomic studies, there is an acute need for better tools that facilitate their analyses against biological functions. To this end, we developed MetQy, an open-source R package designed for query-based analysis of functional units in [meta]genomes and/or sets of genes using the The Kyoto Encyclopedia of Genes and Genomes (KEGG). Furthermore, MetQy contains visualization and analysis tools and facilitates KEGG's flat file manipulation. Thus, MetQy enables better understanding of metabolic capabilities of known genomes or user-specified [meta]genomes by using the available information and can help guide studies in microbial ecology, metabolic engineering and synthetic biology.

**Availability and implementation:** The MetQy R package is freely available and can be downloaded from our group's website (<http://osslab.lifesci.warwick.ac.uk>) or GitHub (<https://github.com/OSS-Lab/MetQy>).

**Contact:** O.Soyer@warwick.ac.uk

### 1 Introduction

The advent of molecular biology has made the characterization and analysis of genomic sequences a key part of all areas of life sciences research. In the case of single-cell organisms, identification of specific functions within the genome directly influences our ability to assess their fitness in a given environment and their potential roles in biotechnology. Particularly, we should theoretically be able to translate genomic data into physiological predictions. Genomic databases are a pre-requisite for making such predictions, but their full use also requires computational tools that allow easy access and systematic analyses of the data.

The Kyoto Encyclopedia of Genes and Genomes (KEGG) is one of the oldest and most comprehensive collections of databases. Its primary aim has been the digitising of current knowledge on genes and molecules and their interactions (Kanehisa, 1997; Kanehisa and Goto, 2000) and it includes 16 databases and 3 sequence data collections (Kanehisa *et al.*, 2017). While these data can be analysed via different tools on the KEGG website, the existing web interface

allows only specific retrieval of information and analyses. Furthermore, although the whole of the data can be downloaded via (paid) FTP access, the systematic analysis of these data in a user-defined manner remains difficult and developing computational analysis tools for this purpose remains a niche expertise that is still not available in many research labs.

There are several specific tools that make use of certain aspects of the KEGG data more available to a wider user-base. Examples include PICRUSt (Langille *et al.*, 2013), BlastKOALA and GhostKOALA (Kanehisa *et al.*, 2016), all of which focus on metagenomics data analysis. However, to our knowledge there are no tools that facilitate the analyses and information retrieval from KEGG with regards to studying the relationship between genomic data and physiological function. Therefore, we have developed MetQy, an open-source, easy-to-use and readily expandable R package for such analyses. MetQy uses the R-platform because it is commonly used among biologists, it is featured in undergraduate education, and it contains extensive statistical packages which are useful in subsequent data analyses.

MetQy was developed to readily interface between the KEGG orthology, module and genome databases and perform automated cross-analyses on them. It consists of a set of functions that allow querying genes, enzymes and functional modules across genomes and vice versa, thereby enabling better understanding of genotype-phenotype mapping in single-celled organisms and providing guidance for cellular engineering in synthetic biology. MetQy can be used 'as-is', since the relevant components of the KEGG databases (downloaded on 20/02/2018) are included within the package. The included KEGG data constitutes only part of the entire encyclopedia and is 'hidden' in the package so that direct access to the data is not possible, complying with KEGG licence. Users with a paid KEGG subscription can use MetQy parsing functions to update the data that the package uses. The MetQy package and GitHub wiki contain extensive documentation and usage examples for each function.

## 2 Software features

MetQy contains three main groups of software functions: data query, parsing and analysis and visualization. These are briefly described below. For more detailed information and usage examples, please see the package documentation and GitHub wiki.

### 2.1 Metabolic query functions

The *query* family of functions allows the user to query the KEGG data structures in a systematic (and automated) way. Users without FTP access can analyse the KEGG genome, module and ortholog databases indirectly by using this family of functions on built-in formatted KEGG data which is not directly accessible by the user. Additionally, these functions feature optional arguments that allow users to provide up-to-date data (by using the *parsing* functions on KEGG FTP data) or their own data structures, such as custom-made KEGG-style modules. Additional query functions can be readily developed by the users, allowing expansion of MetQy. MetQy features five query functions for key functional analyses.

*query\_genomes\_to\_modules* calculates the module completeness fraction (*mcf*) given a set of genes or genomes. It returns a matrix showing the *mcf* for each module. The *mcf* calculation is based on block-based, logical KEGG module definition (see GitHub wiki). The function input is the modules to be queried (default is all KEGG modules) and the set of genes to be considered. The gene set can be provided either as a set of KEGG ortholog or Enzyme Commission (EC) numbers, or as genome identifier(s), with the latter case resulting in automatic retrieval of all genes for the genome(s).

While the implementation of *query\_genomes\_to\_modules* function is similar to KEGG mapper [a web interface tool that performs a similar task (<http://www.genome.jp/kegg/mapper.html>; Kanehisa et al., 2017)], there are several key features that are different. The KEGG Mapper's web interface does not allow for module-specific evaluation nor for automation of the analysis. Our implementation allows for specific KEGG modules to be evaluated, given their ID, name and/or class. It also provides the capacity to determine the *mcf* of a module, rather than only identifying modules that are complete or that have one block missing. Finally, as EC numbers are widely used in systems biology, we used the KEGG orthology to translate the K number-based module definitions to EC number-based module definitions. This allows for module evaluation based on both K and EC numbers.

*query\_module\_to\_genomes* determines the KEGG genome(s) that have user-specified module(s) that are complete above a *mcf* threshold (defaults to 1, i.e. complete). *query\_gene\_to\_modules* determines those KEGG modules that feature specific user-specified gene(s).

*query\_genes\_to\_genomes* determines which KEGG genomes contain user-specified gene(s). *query\_missingGenes\_from\_module* determines the missing gene(s) (K or EC numbers) that would be required to have a complete KEGG module within a genome (or gene set).

### 2.2 Parsing KEGG databases

MetQy comes with built-in data components of KEGG. It is, however, possible for users with FTP KEGG access to update these data components to their latest version. The MetQy *parsing* functions allow the production of the updated data, by formatting the relevant KEGG data files into R structures. They can also be used as stand-alone functions to introduce KEGG data into the R environment. All *query* functions have been designed to take in these updated data.

MetQy features two generic parsing functions that deal with the two main KEGG file types: files without extension (*parseKEGG\_file*) and '.list' files (*parseKEGG\_file.list*). *parseKEGG\_file.list* formats KEGG files containing a mapping between two KEGG database entries into binary matrices. For example, the mapping between K numbers and EC numbers is contained in the 'ko\_enzyme.list' file and shows which K numbers correspond to which EC numbers. *parseKEGG\_file* formats a KEGG database file into an R data frame by automatically detecting fields of the KEGG data and transforms these into variables. MetQy also contains file-specific functions that use these generic functions.

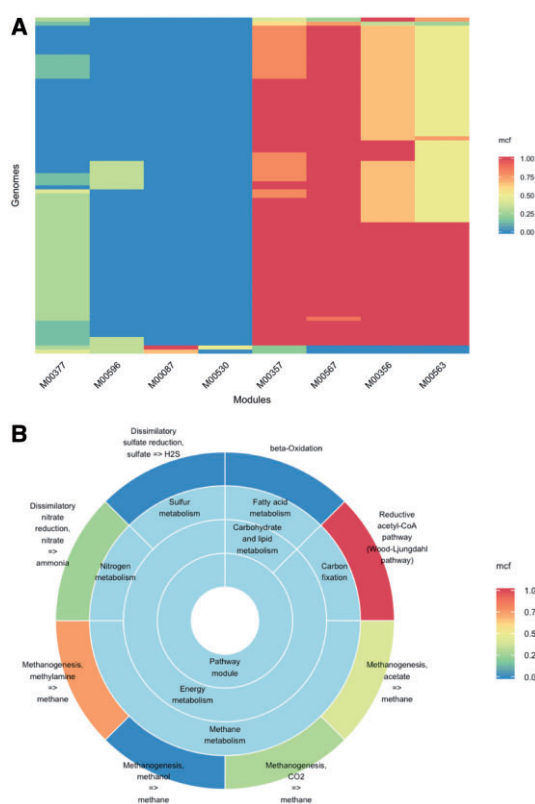
### 2.3 Analysis and visualization

The analysis and visualization family of functions are designed to facilitate the analysis primarily of the output of the *query\_genomes\_to\_modules* function, which generates a matrix of *mcf* values for the genomes and modules analysed. There are three *analysis* and five *plot* (visualization) functions.

*analysis\_pca\_mean\_distance\_calculation* is designed to process the output of a principal component analysis (PCA) performed on the *mcf* matrix (this can be done for example by applying the R function *stats::prcomp* function). It uses the resulting numeric matrix containing the principal components to calculate the mean Euclidean distance as a measure of spread or variation (of the data). This assumes that every row represents a multi-dimensional point (a genome in this case), with coordinates given in the corresponding columns. The mean Euclidean distance of *p* points is calculated by adding the computed pairwise Euclidean distance in *n* dimensions between all the points divided by the total number of distances.

*analysis\_pca\_mean\_distance\_grouping* takes in the numeric matrix resulting from performing a PCA on the *mcf* matrix and a factor, such as genus, to group the rows (genomes) of the matrix and uses the previous function (*analysis\_pca\_mean\_distance\_calculation*) to calculate the mean Euclidean distance for each group.

*analysis\_genomes\_module\_output* takes in the *mcf* matrix (genomes and modules as rows and columns, respectively) and produces a series of analyses and generates a report automatically by default. These analyses comprise of: (i) reporting the number of genomes (data sets) and modules analysed, producing a (ii) heatmap of the *mcf* of all genomes and modules analysed, (iii) a boxplot of the *mcf* across all genomes for each module, (iv) a scatter plot of the SD of the *mcf* across all genomes for each module and (v) identifying any modules that have a constant (zero-variance) *mcf* across all genomes and producing a table. In addition, the function performs, for every factor group specified, the following analyses: (vi) group the genomes according to that factor and create a heatmap of the mean *mcf* for each module across the genomes that make up each group, (vii) carry out a PCA analysis on all the *mcf* data, showing



**Fig. 1.** Visualization of some of the results obtained from an example analysis (Section 3.1). **(A)** Heatmap representation of module fraction completeness (*mcf*) across selected genomes (y-axis) and modules (x-axis). The *mcf* value is colour-coded as per the provided mapping scheme shown. **(B)** A sunburst diagram showing the *mcf* of different modules and their functional classes as obtained from the analysis of a specific genome (genome ID: T04272). The *mcf* value is colour-coded as per the provided mapping scheme shown. The data for both plots was obtained using MetQy function 'query\_genomes\_to\_modules'.

the cumulative variance and generating a PC plot, (viii) visualize the PC plot with an overlay of the factor grouping and, finally, (ix) measure the within-group (per factor) variance, using the mean Euclidean distance as a proxy for spread.

*plot\_heatmap* can be used to visualize the *mcf* calculated by the *query\_genomes\_to\_modules* function as a colour mapped matrix (with genomes against modules). *plot\_scatter\_byFactors* allows the automatic grouping of data as determined by a factor and produces a scatter plot with groups overlaid by colour. *plot\_scatter* is useful to visualize numerical data associated to data groups generated by a factor. This category-based visualization can be used to plot the SD for each module's *mcf* or the mean Euclidean distance (see the analysis description above for more details). *plot\_variance\_boxplot* takes the *mcf* matrix and produces a boxplot for each module. *plot\_sunburst* makes a hierarchical arrangement of categorical data, such as KEGG module classes, and represents it in a dart-style, where the inner ring contains the most general (highest level) information which can be divided into sub-categories (rings going outwards). The final ring represents the most specific level of information and can be coloured by either the counts of the data or an additional set of values provided by the user (refer to the GitHub wiki for more information).

### 3 Uses and applications

MetQy facilitates the general usability of the KEGG database and allows users to gain qualitative information about the functional capacity of a given organism or gene set. Anticipated uses of the tool include synthetic biology, where it can facilitate the design and guiding of metabolic engineering studies by identifying missing genes needed for an organism to have a complete KEGG module, and identifying KEGG genomes with desired metabolic capabilities. For systems biology applications, it allows identification of key physiological features of organisms and development of stoichiometric metabolic models by analysing module completeness in specific genomes and identifying transporter modules and carbon utilization routes in genomes. Finally, in microbial ecology, MetQy can allow species-function mappings in metagenomes and insights into functional capabilities of ecological groups by analysing the metabolic capacity of novel genomes from metagenomic studies. Organisms can be put into different functional groups, and the functional profiles of different environments compared.

#### 3.1 Example of usage

To demonstrate some possible uses of MetQy functions, we have included a coded example on the MetQy GitHub wiki pages. This example demonstrates how MetQy can be used to retrieve KEGG genome data and how the metabolic functions of the extracted/matched organisms can be queried/identified in terms of KEGG modules. In the presented example, we evaluate the module completeness fraction (*mcf*) in methanogen genomes, focusing on sample KEGG modules loosely relating to the anaerobic digestion process (note that any user specified modules, or all KEGG can be used in a real analysis). We then visualize the results of this analysis as a heatmap using MetQy function *plot\_heatmap* (Fig. 1A). In this example case, this analysis highlighted a specific module that is expected to be essential for methanogenesis (M00567: Methanogenesis, CO<sub>2</sub> => methane) and that was almost fully complete in most genomes as expected (*mcf* >= 0.75 in 96% of genomes), but incomplete in some genomes. This prompted us to analyse the genomes that had a lower *mcf* for this key module. We thus identified the genome T04272 (Methanogenic archaeon ISO4-H5) as an interesting methanogen to focus on and used another MetQy function *plot\_sunburst* to analyse all of its modules' *mcf* through a sunburst plot (Fig. 1B). Furthermore, we identified the genes that were missing for that module to be complete (for that organism).

While this example highlights how specific MetQy functions can be utilized on their own to develop a specific analysis pipeline, it is also possible to use MetQy functions to perform an automated analysis on a set of genomes grouped by genus (or another grouping factor provided by the user, e.g. species or sample origin) and generate a comprehensive report in an automated fashion (see description for *analysis\_genomes\_module\_output* function, the PDF report file in the GitHub repository and the worked-out example in the GitHub wiki).

### Acknowledgement

The authors acknowledge Sean Aller for helpful comments and David Selby for sharing his expertise in developing R packages.

### Funding

This work is funded by The University of Warwick and by the Biotechnological and Biological and Engineering and Physical Sciences Research Councils (BB- and EPSRC), with grant IDs: EP/L016494/1 (to

the Centre for Doctoral Training in Synthetic Biology, SynBioCDT), BB/K003240/2 (to OSS), BB/M017982/1 (to the Warwick Integrative Synthetic Biology Centre, WISB).

*Conflict of Interest:* none declared.

## References

- Kanehisa,M. (1997) A database for post-genome analysis. *Trends Genet.*, **13**, 375–376.
- Kanehisa,M. and Goto,S. (2000) KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **28**, 27–30.
- Kanehisa,M. et al. (2016) BlastKOALA and GhostKOALA: KEGG tools for functional characterization of genome and metagenome sequences. *J. Mol. Biol.*, **428**, 726–731.
- Kanehisa,M. et al. (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.*, **45**, D353–D361.
- Langille,M.G.I. et al. (2013) Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. *Nat. Biotechnol.*, **31**, 814–821.

## E.2 MetQy package documentation

The MetQy package documentation has been included in this Appendix section and is available at:

[https://github.com/OSS-Lab/MetQy/blob/master/MetQy\\_1.1.0.pdf](https://github.com/OSS-Lab/MetQy/blob/master/MetQy_1.1.0.pdf)

# Package ‘MetQy’

May 24, 2018

**Title** Metabolic Analysis using Queries

**Version** 1.1.0

**BugReports** <https://github.com/OSS-Lab/MetQy/issues>

**Description** MetQy facilitates analysis and data mining of KEGG. The package consists of several families of functions. 'parseKEGG' functions allow for easy reading and formatting of the KEGG databases. 'query' functions carry out metabolic analysis given genes or genomes (both user-specified or from the KEGG database) and KEGG module search terms. 'plot' and 'analysis' functions facilitate data visualisation and analyses such as module variance and PCA. 'misc' functions support the other function families.

**Depends** R (>= 3.4.1)

**Imports** dplyr,ggplot2,gsubfn,reshape2,xtable

**License** University of Warwick non commercial use licence (see LICENCE  
file: <https://github.com/OSS-Lab/MetQy/blob/master/LICENCE>)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Suggests** testthat

**Author** Andrea Martinez-Vernon [aut, cre],  
Soyer Orkun [ctb],  
Frederick Farrell [ctb]

**Maintainer** Andrea Martinez-Vernon <asmvernon@gmail.com>

## R topics documented:

analysis_genomes_module_output . . . . .	2
analysis_pca_mean_distance_calculation . . . . .	4
analysis_pca_mean_distance_grouping . . . . .	5
data_example_ECnumbers_vector . . . . .	6
data_example_genomeIDs . . . . .	7
data_example_KOnumbers_vector . . . . .	7
data_example_moduleIDs . . . . .	8
data_example_multi_EC_KOs . . . . .	9
data_example_sunburst . . . . .	10
data_example_sunburst_fill_by . . . . .	10
data_module_shortcode_mapping . . . . .	11

<code>misc_axisRound</code> . . . . .	12
<code>misc_check_duplicate_names</code> . . . . .	12
<code>misc_create_labels</code> . . . . .	13
<code>misc_evaluate_block</code> . . . . .	13
<code>misc_geneVector_module</code> . . . . .	14
<code>misc_module_definition_block_EC</code> . . . . .	16
<code>misc_module_definition_check</code> . . . . .	16
<code>misc_module_definition_optional</code> . . . . .	17
<code>misc_module_subgroup_indexing</code> . . . . .	18
<code>parseKEGG_compound</code> . . . . .	18
<code>parseKEGG_enzyme</code> . . . . .	19
<code>parseKEGG_execute_all</code> . . . . .	21
<code>parseKEGG_file</code> . . . . .	22
<code>parseKEGG_file.list</code> . . . . .	23
<code>parseKEGG_genome</code> . . . . .	23
<code>parseKEGG_ko</code> . . . . .	25
<code>parseKEGG_ko_enzyme</code> . . . . .	26
<code>parseKEGG_ko_reaction</code> . . . . .	27
<code>parseKEGG_module</code> . . . . .	28
<code>parseKEGG_process_KEGG_taxonomy</code> . . . . .	30
<code>parseKEGG_reaction</code> . . . . .	31
<code>plot_heatmap</code> . . . . .	33
<code>plot_scatter</code> . . . . .	34
<code>plot_scatter_byFactors</code> . . . . .	35
<code>plot_sunburst</code> . . . . .	37
<code>plot_variance_boxplot</code> . . . . .	39
<code>query_genes_to_genomes</code> . . . . .	40
<code>query_genes_to_modules</code> . . . . .	41
<code>query_genomes_to_modules</code> . . . . .	42
<code>query_missingGenes_from_module</code> . . . . .	45
<code>query_modules_to_genomes</code> . . . . .	46

<b>Index</b>	<b>48</b>
--------------	-----------

---

<code>analysis_genomes_module_output</code>	<i>Process the output generated by <code>query_genomes_to_modules()</code>.</i>
---	---

---

## Description

Process the output generated by `query_genomes_to_modules()`.

## Usage

```
analysis_genomes_module_output(FRACTION_MATRIX, QUERIES = NULL,
  outPath = "", report_file = "report.tex", figType = c(".eps", ".png"),
  FACTOR = NULL, ...)
```

## Arguments

FRACTION_MATRIX	- matrix. \$MATRIX matrix generated by query_genomes_to_modules().
QUERIES	- optional. \$QUERIES data frame generated by query_genomes_to_modules() containing the search query performed.
outPath	- optional. String to indicate path to store output file and figures. Default (Sys.Date() as 'YYYY_MM_DD', e.g. '2000_01_31').
report_file	- optional. File name for LaTeX report (see Details for path handling). Default ("report.tex" saved to outPath; use NULL to suppress report generation).
figType	- optional. Character vector to indicate file extension for figures. Default (c(".eps", ".png")).
FACTOR	- optional. Character vector or list of character vectors indicating the grouping factor of the datasets. Default (NULL). See Details.
...	- further arguments for analysis_pca_mean_distance_grouping or argument factor_labs for plot_scatter_byFactors.

## Details

Below are the steps carried on the query\_genomes\_to\_modules() output.

(1) Module fraction completeness (mfc) visualization of the query\_genomes\_to\_modules() output; generates file: module\_allOrgs.figType.

(2) variance analysis to identify complete and absent modules, as well as modules with zero variance (i.e. same fraction present across all modules); generates files: module\_allOrgs\_sd.figType, module\_allOrgs\_sd\_boxplot.figType and module\_constant\_presence.txt.

(3) a principal component analysis (PCA; using the prcomp from the 'stats' package) is done and a plot of the cumulative variance and the first to principal components are generated; generates files in PCA/: pca\_plot.figType, pca\_sd.figType and pca.rda

When FACTOR is specified, a mean distance is calculated as a proxy for inner-group spread (using as many dimensions as specified by the optional additional argument nDim; used by analysis\_pca\_mean\_distance\_grouping()). This generates additional files: module\_mean\_dist\_output\_FACTOR.rda and the following for each factor (multiple factors possible if FACTOR is a list):

module\_<FACTOR>\_mean.png, module\_<FACTOR>\_sd.figType,

PCA/plot\_mean\_dist\_<FACTOR>.figType and PCA/plot\_scatter\_<FACTOR>.figType

It will be assumed that if report\_file has a folder structure (identified by "\") a path has been given. Otherwise, the report will be written to outPath. If report\_file is set to NULL, the report is not generated.

## Value

This function does not return anything, but does generate multiple files (see Details). This function automatically generates a ".tex" file (LaTeX report), which can be externally processed to generate a PDF file, unless report\_file is set to NULL.

ggplot objects for all figures are generated and saved in an R object ("module\_output\_plots.rda") within the output directory.

## See Also

query\_genomes\_to\_modules, analysis\_pca\_mean\_distance\_grouping, plot\_heatmap

---

analysis\_pca\_mean\_distance\_calculation

*Given a set of coordinates, calculate the mean distance between all points.*

---

## Description

Given a set of coordinates, calculate the mean distance between all points.

## Usage

```
analysis_pca_mean_distance_calculation(MATRIX, ...)
```

## Arguments

MATRIX	- matrix with the rows referring to the points with N columns containing the coordinates (and therefore N dimensions).
...	- further arguments (currently unsupported)

## Details

The mean distance of  $p$  points is calculated by the sum of the individual Euclidean distances divided by the total number of distances (given by  $p * (p - 1) / 2$ ).

## Value

The mean distance (numeric).

## Examples

```
data(data_example_moduleIDs)
data(data_example_genomeIDs)

# Calculate the module completion fraction (mcf) for the genomes
# and modules contained in the data objects above.
OUT <- query_genomes_to_modules(data_example_genomeIDs,
                                MODULE_ID = data_example_moduleIDs)

pca <- prcomp(OUT$MATRIX)

mean_dist <- analysis_pca_mean_distance_calculation(pca$x)
# [1] 0.4805169
```



---

analysis\_pca\_mean\_distance\_grouping

*Given a set of coordinates and a grouping variable, the mean distance is calculated for each group.*

---

## Description

Given a set of coordinates and a grouping variable, the mean distance is calculated for each group.

## Usage

```
analysis_pca_mean_distance_grouping(MATRIX, FACTOR, factor_labs = NULL,  
  nDim = 2, plot_mean_dist = T, Filename = "plot_mean_dist.pdf",  
  plot_top_percent = NULL, ...)
```

## Arguments

MATRIX	- matrix with the point coordinates (two column minimum).
FACTOR	- used to split the data into groups. Character vector (or list of vectors) with the same length as rows in MATRIX.
factor_labs	- optional. Character vector of the length of FACTOR if FACTOR is a list or length 1 if FACTOR is a character vector. Default (NULL; FACTOR names or letters used).
nDim	- optional. Numeric vector of dimensions to use for the point coordinates. Default (2).
plot_mean_dist	- logical. Should a plot of the mean distances be generated? Default (TRUE).
Filename	- optional. Filename with path and extension. String added to distinguish FACTOR groups. Default ("plot_mean_dist.pdf" saved to working directory).
plot_top_percent	- optional. Numeric vector between 0 and 1 or integer. Default (NULL, i.e. not plotted).
...	- further arguments for plot_scatter().

## Details

The mean distance of groups with one member (derived from the FACTOR labels) cannot be calculated.

nDim must be larger than 2.

plot\_top\_percent refers to the fraction or number of groups with the highest calculated mean distance that should be plotted on a scatter plot. Items are recycled if not as many as the number of grouping factors is provided.

## Value

List containing a table of the mean distances (columns for group name, mean distance and size of group (N)) and a ggplot object of the data (NAs removed) if plot\_mean\_dist is TRUE.

## See Also

analysis\_pca\_mean\_distance\_calculation

## Examples

```
data(data_example_moduleIDs)
data(data_example_genomeIDs)

# Calculate the module completion fraction (mcf) for the genomes
#           and modules contained in the data objects above.
OUT      <- query_genomes_to_modules(data_example_genomeIDs,
                                     MODULE_ID = data_example_moduleIDs)

pca <- prcomp(OUT$MATRIX)

# Group data
this_FACTOR <- rep(LETTERS[1:5], length(data_example_genomeIDs)/5)
mean_dist_output <- analysis_pca_mean_distance_grouping(pca$x, this_FACTOR, xLabs_angle = F,
                                                         Width = 2, Height = 1.5,
                                                         Filename = "plot_pca_scatter.png")
```

---

data\_example\_ECnumbers\_vector

*Example of a vector containing Enzyme Commission (EC) numbers.*

---

## Description

A dataset containing an example of a vector containing Enzyme Commission (EC) numbers that can be used with the function `misc_geneVector_module()` or by the `query_genomes_to_modules()` function after formatting it into a data frame (see function description).

## Usage

```
data_example_ECnumbers_vector
```

## Format

A character vector with 568 entries

**ECs** 1.1.1.10, 1.1.1.102, 1.1.1.105, 1.1.1.12, 1.1.1.14, 1.1.1.153, ...

## Details

When generating an EC number vector, make sure that all entries have the 4 nomenclature positions ("-" denotes an unspecified field, e.g. "1.1.1.-").

## Source

[http://www.kegg.jp/kegg-bin/get\\_htext?eco00001](http://www.kegg.jp/kegg-bin/get_htext?eco00001) and expand the subsections to see the ECs.

## See Also

```
data_example_KOnumbers_vector
```

## Examples

```
# Load data
data("data_example_ECnumbers_vector")
head(data_example_ECnumbers_vector)
# [1] "1.1.1.10" "1.1.1.102" "1.1.1.105" "1.1.1.12" "1.1.1.14" "1.1.1.153"
```

---

data\_example\_genomeIDs

*Example of vector containing KEGG genome IDs to be used by query\_genomes\_to\_modules().*

---

## Description

A dataset containing an example of a character vector that can be used as the GENOME\_INFO input to the function query\_genomes\_to\_modules().

## Usage

```
data_example_genomeIDs
```

## Format

An object of class character of length 25.

## See Also

plot\_heatmap

## Examples

```
# Load data
data("data_example_genomeIDs")
head(data_example_genomeIDs)
# [1] "T04503" "T04203" "T03253" "T00526" "T00341" "T00552"
```

---

data\_example\_K0numbers\_vector

*Example of a vector containing KEGG Ortholog (KO) identifiers*

---

## Description

A dataset containing an example of a vector containing KEGG Orthologs (K numbers or KOs) that can be used with the function misc\_geneVector\_module() or by the function query\_genomes\_to\_modules() after formatting it into a data frame (see function description).

## Usage

```
data_example_K0numbers_vector
```

## Format

A character vector with 2896 entries

**KOs** K00005, K00009, K00012, K00013, K00014, ...

## Source

[http://www.kegg.jp/kegg-bin/get\\_htext?eco00001](http://www.kegg.jp/kegg-bin/get_htext?eco00001) and expand the subsections to see the KOs.

## See Also

data\_example\_ECnumbers\_vector

## Examples

```
# Load data
data("data_example_KOnumbers_vector")
head(data_example_KOnumbers_vector)
# [1] "K00005" "K00009" "K00012" "K00013" "K00014" "K00024"
```

---

data\_example\_moduleIDs

*Example of vector containing KEGG module IDs to be used by query\_genomes\_to\_modules().*

---

## Description

A dataset containing an example of a character vector that can be used as the MODULE\_ID input to the function query\_genomes\_to\_modules().

## Usage

data\_example\_moduleIDs

## Format

An object of class character of length 6.

## See Also

plot\_heatmap

## Examples

```
# Load data
data("data_example_moduleIDs")
head(data_example_moduleIDs)
# [1] "M00356" "M00357" "M00563" "M00567" "M00596" "M00377"
```

---

`data_example_multi_EC_KOs`

*Example of a data frame with multiple datasets.*

---

## Description

A data frame containing an example of multiple datasets that could be analysed with the function `misc_geneVector_module()` or by the function `query_genomes_to_modules()` after formatting it into a data frame (see function description). The dataset entries are to illustrate what the KEGG genome data looks like.

## Usage

`data_example_multi_EC_KOs`

## Format

A data frame with the following columns:

ID	- KEGG genome ID, T number (e.g. "T00001")
ORG_ID	- KEGG organism ID, 3-4 letter code (e.g. "eco")
ORGANISM	- Organism name (e.g. "Escherichia coli K-12 MG1655")
KOs	- Concatenated string with the K numbers (e.g. "K00013;K00014;K00018;...").
ECs	- Concatenated string with the EC numbers (e.g. "1.1.1.1;1.1.1.100;1.1.1.130;...")

## Details

The columns 'KOs' or 'ECs' need to be pointed at to do the module mapping and is specified by using the argument 'mapBy' in `query_genomes_to_modules()` and The default is to use the column named 'KOs', but it can be named differently and specified in the argument field.

Specify the character to be used to split the string with the K/EC numbers using the argument 'split\_vector\_by' in `query_genomes_to_modules()`

## See Also

`query_genomes_to_modules`

## Examples

```
# Load data
data("data_example_multi_EC_KOs")
head(data_example_multi_EC_KOs)
```

---

`data_example_sunburst` *Example of the hierarchical data needed to generate a sunburst plot.*

---

### Description

A dataset containing an example of a data frame that can be used with the function `plot_sunburst()`. This function helps visualize hierarchical information by making concentric "donut" plots which has the highest hierarchical level data at the centre going out.

### Usage

```
data_example_sunburst
```

### Format

An object of class `data.frame` with 8 rows and 4 columns.

### See Also

`plot_sunburst`

### Examples

```
# Load data
data("data_example_sunburst")
head(data_example_sunburst)
#      CLASS_I CLASS_II CLASS_III NAME_SHORT
# 1 Pathway module Carbohydrate & lipid metabolism Fatty acid metabolism beta-Oxidation
# 2 Pathway module Energy metabolism Methane metabolism Methanogenesis, from methanol
# 3 Pathway module Energy metabolism Methane metabolism Methanogenesis, from acetate
# 4 Pathway module Energy metabolism Carbon fixation Reductive acetyl-CoA pathway
# 5 Pathway module Energy metabolism Nitrogen metabolism Dissimilatory nitrate reduction
# 6 Pathway module Energy metabolism Methane metabolism Methanogenesis, from methylamine
```

---

`data_example_sunburst_fill_by`  
*Example of vector to provide alternative values to fill the outermost level of a sunburst plot.*

---

### Description

A dataset containing an example of a numerical vector that can be used as the `fill_by` input to the function `plot_sunburst()`.

### Usage

```
data_example_sunburst_fill_by
```

## Format

An object of class `numeric` of length 8.

## See Also

`plot_sunburst`

## Examples

```
# Load data
data("data_example_sunburst_fill_by")
head(data_example_sunburst_fill_by)
# M00087 M00356 M00357 M00377 M00530 M00563
# 0.0 1.0 0.8 0.0 0.0 0.5
```

---

`data_module_shortName_mapping`

*Data frame used to populate the short name in module\_reference\_table*

---

## Description

This data frame contains the following columns and is used to provide a manually abbreviated name to the modules to ease plotting.

## Usage

```
data_module_shortName_mapping
```

## Format

A data frame with the following columns:

ID	- KEGG module ID, M number (e.g. "M00001")
NAME	- KEGG module name
NAME_SHORT	- Manually abbreviated KEGG module name (unique entries)

## See Also

`parseKEGG_module`

## Examples

```
# Load data
data("module_shortName_mapping")
head(module_shortName_mapping)
```

---

<code>misc_axisRound</code>	<i>Find best value to round axis to.</i>
-----------------------------	--

---

### Description

Find best value to round axis to.

### Usage

```
misc_axisRound(Vector, roundBy = NULL, Min = NULL, ...)
```

### Arguments

<code>Vector</code>	- values that are being plotted.
<code>roundBy</code>	- optional. The value to use for rounding. Default (NULL).
<code>Min</code>	- optional. The value to use as the starting value for the axis. Default (NULL).
<code>...</code>	- further arguments (currently unsupported)

### Value

List containing the minimum and maximum axis values (`$min` and `$max`, respectively) and the array that can be used to indicate the axis breaks (`$array`)

---

<code>misc_check_duplicate_names</code>	<i>Check for duplicated strings and append letter code to distinguish them.</i>
---	---

---

### Description

Check for duplicated strings and append letter code to distinguish them.

### Usage

```
misc_check_duplicate_names(NAME_VECTOR, ...)
```

### Arguments

<code>NAME_VECTOR</code>	- character vector.
<code>...</code>	- further arguments (currently unsupported)



---

<code>misc_create_labels</code>	<i>Creates labels to add to a character vector to make all entries unique</i>
---------------------------------	---

---

### Description

Creates a label vector of length N, such that they are unique and help distinguish them

### Usage

```
misc_create_labels(N, ...)
```

### Arguments

N	- numeric. Indicates the length of the label vector to be return.
...	- further arguments (currently unsupported)

### Value

A character vector of length N with unique set of alpha-numeric labels.

---

<code>misc_evaluate_block</code>	<i>Evaluate a KEGG module block definition.</i>
----------------------------------	---

---

### Description

Evaluate a KEGG module block definition given a vector of genes (KEGG Orthologs -KOs-, identified with K number). Some KOs can be mapped to Enzyme Commission (EC) numbers.

### Usage

```
misc_evaluate_block(gene_vector, BLOCK, KO_in_DEF_EC = FALSE, ...)
```

### Arguments

gene_vector	- vector containing either K or EC numbers. See Details.
BLOCK	- KEGG module DEFINITION BLOCK. See <code>parseKEGG_module</code> .
KO_in_DEF_EC	- logical. If enzyme IDs are given, should lingering K numbers in the module DEFINITION be assumed to be present? Default (FALSE).
...	- further arguments (currently unsupported)

### Details

gene\_vector must contain either K or EC numbers.

---

`misc_geneVector_module`

*Map a list of EC or K numbers to KEGG modules.*

---

## Description

This function maps the list of Enzyme Commission (EC) numbers or KEGG orthologs (specified as K numbers) given by `gene_vector` to the KEGG modules using an in-built reference table (`module_reference_table`) of all the KEGG modules in the database at the time of release.

## Usage

```
misc_geneVector_module(gene_vector, MODULE_ID, SEARCH_NAME, SEARCH_CLASS_I,  
  SEARCH_CLASS_II, SEARCH_CLASS_III, EXCLUDE_NAME,  
  use_module_reference_table = NULL, ...)
```

## Arguments

<code>gene_vector</code>	- character vector listing EC or K numbers. If more K numbers are given, the K number-based definition will be used. See Value.
<code>MODULE_ID</code>	- optional. Character vector listing specific module IDs (e.g. M00001).
<code>SEARCH_NAME</code>	- optional. Character vector listing terms to search in NAME field (case-insensitive).
<code>SEARCH_CLASS_I</code>	- optional. Character vector listing terms to search in CLASS_I field (case-insensitive).
<code>SEARCH_CLASS_II</code>	- optional. Character vector listing terms to search in CLASS_II field (case-insensitive).
<code>SEARCH_CLASS_III</code>	- optional. Character vector listing terms to search in CLASS_III field (case-insensitive).
<code>EXCLUDE_NAME</code>	- optional. Character vector listing terms that if matched in NAME field will be excluded (case-insensitive).
<code>use_module_reference_table</code>	- optional. Provide a data frame with updated KEGG module database OR with custom-made modules. Default (NULL; inbuilt data used). See Details.
<code>...</code>	- further arguments such as <code>K0_in_DEF_EC</code> . See Details.

## Details

The modules to be analysed are determined by search queries. These can search across the module NAME, CLASS (I-III) or by specifying the module ID (M number). See the Argument section. When none of the optional arguments are specified, the default is to search all modules. An additional argument, `EXCLUDE_NAME`, can be used to exclude modules with a certain term in the NAME field. For example, specifying `EXCLUDE_NAME = "biosynthesis"` would return the search across all modules, except those that contain "biosynthesis" in the name. For instance, module "M00005" named "PRPP biosynthesis, ribose 5P => PRPP" and 168 others would be excluded from the search. If neither `MODULE_ID` or at least one `SEARCH` term is specified, the default is to analyse all modules.

The `use_` argument allow users with KEGG FTP access to provide the updated data from the KEGG databases in the form of reference tables AND/OR for advanced users to provide custom-made

modules (see below). These reference tables can be generated with `parseKEGG_module` and need to have a specific format (see function descriptions for details on format).

The module definition (contained in `module_reference_table`) describes the relationship between genes and modules and is used to identify the modules in which gene is involved. The user can provide custom-made module definitions that use the logical expression format (however, the table format must be conserved!).

## Value

List: `$FRACTION` matrix containing the completeness fraction for the modules matched in the query along the columns.

`$METADATA` data frame with the metadata for the modules matched in the query. Columns:

- |         |                                |   |
|---------|--------------------------------|---|
| (1)     | <code>MODULE_ID</code>         | - M number (e.g. M00001);   |
| (2)     | <code>MODULE_NAME</code>       | - the KEGG given name for the module;   |
| (3)     | <code>MODULE_NAME_SHORT</code> | - manually shortened module name (for plotting purposes);   |
| (4 - 6) | <code>CLASS_I - III</code>     | - hierarchical module classes;  |
| (7)     | <code>DEFINITION</code>        | - KEGG module definition in terms of K or EC numbers (without optional K or EC numbers);                            |
| (8)     | <code>OPTIONAL</code>          | - the optional K or EC numbers that are part of the KEGG module definition (NA otherwise);                          |
| (9)     | <code>KOs_IN_DEF</code>        | - logical flag indicating whether there are K numbers involved in that module definition (if EC numbers are given); |

`$ADD_INFO` data frame with additional output information. Columns:

- |     |                                |   |
|-----|--------------------------------|---|
| (1) | <code>MODULE_ID</code>         | - M number (e.g. M00001);   |
| (2) | <code>MODULE_NAME_SHORT</code> | - manually shortened module name (for plotting purposes);                           |
| (3) | <code>FRACTION</code>          | - the number of complete blocks divided by the total number of blocks;              |
| (4) | <code>nBLOCKS</code>           | - the number of blocks that make up the KEGG module;                                |
| (6) | <code>COVERAGE</code>          | - the K numbers or ECs that are present and involved in the KEGG module definition; |
| (7) | <code>OPTIONAL_PRESENT</code>  | - the K numbers or ECs that are present and are listed as optional;                 |

See `parseKEGG_module` or visit <http://www.genome.jp/kegg/module.html> for more information.

## See Also

`parseKEGG_module`

## Examples

```
output_module_EC<- misc_geneVector_module(data_exampleECnumbers_vector,
                                           MODULE_ID=paste("M0000",1:5,sep=""))

output_module_KOs<- misc_geneVector_module(data_exampleKOnumbers_vector,
                                           MODULE_ID=paste("M0000",1:5,sep=""))
```

---

`misc_module_definition_block_EC`

*KEGG module definition processing: subblocks*

---

### Description

Format the KEGG module database - Format the KEGG module DEFINITION for ease of analysis by excluding optional KEGG orthologs. Used by `parseKEGG_module()`.

### Usage

`misc_module_definition_block_EC(BLOCK, ORTHOLOGS, ...)`

### Arguments

<code>BLOCK</code>	- string containing a block of a KEGG module DEFINITION (logical expression using K numbers).
<code>ORTHOLOGS</code>	- vector listing the K numbers and related EC numbers.
<code>...</code>	- further arguments (currently unsupported)

### Details

The KEGG module definition uses optional KEGG orthologs, indicated by a "-". These are removed and the corresponding EC number stored.

### See Also

`parseKEGG_module`

---

`misc_module_definition_check`

*KEGG module definition processing: block formatting.*

---

### Description

Format the KEGG module DEFINITION for ease of analysis.

### Usage

`misc_module_definition_check(DEFINITION, ...)`

### Arguments

<code>DEFINITION</code>	- string containing a KEGG module DEFINITION (logical expression using K numbers).
<code>...</code>	- further arguments (currently unsupported)

## Details

The KEGG module definition uses both spaces and plus signs to indicate 'AND' operations. However, the 'AND' operation can be used to split the module definition into BLOCKS or to indicate molecular complex composition. To simplify analysis, we will use spaces ONLY to delimit KEGG module BLOCKS and the plus sign ONLY to indicate molecular complexes and 'AND' operations within blocks.

## See Also

parseKEGG\_module

---

misc\_module\_definition\_optional

*KEGG module definition processing: optional KEGG ortholog exclusion*

---

## Description

Format the KEGG module database - Format the KEGG module definition for ease of analysis by excluding optional KEGG orthologs. Used by parseKEGG\_module().

## Usage

```
misc_module_definition_optional(BLOCK, ORTHOLOGS, ...)
```

## Arguments

BLOCK	- string containing a block of a KEGG module DEFINITION (logical expression using K numbers).
ORTHOLOGS	- vector listing the K numbers and related EC numbers.
...	- further arguments (currently unsupported)

## Details

The KEGG module definition uses optional KEGG orthologs, indicated by a "-". These are removed and the corresponding EC number stored.

## Value

A list with the formatted block (`$thisBlock`) and any optional K and EC numbers (`$thisOptional_K0` and `$thisOptional_EC`)

## See Also

parseKEGG\_module

---

`misc_module_subgroup_indexing`

*Subgroup check - formatting KEGG module definition.*

---

### Description

This functions helps format the KEGG module DEFINITION by checking all the bracket-delimited BLOCKS to then remove flanking brackets (unnecessary).

### Usage

```
misc_module_subgroup_indexing(DEFINITION, ...)
```

### Arguments

DEFINITION	- KEGG module definition
...	- further arguments (currently unsupported)

### See Also

`parseKEGG_module`, `misc_module_definition_check`

---

`parseKEGG_compound`

*Parse the KEGG compound database*

---

### Description

Read the KEGG compound database text file and format it into a reference table.

### Usage

```
parseKEGG_compound(KEGG_path, outDir = "output", verbose = T, ...)
```

### Arguments

KEGG_path	- string pointing to the location of the KEGG database parent folder. The path to the required file is contained within the function.
outDir	- string pointing to the output folder. Default ("output/").
verbose	- logical. Should progress be printed to the screen? Default (TRUE).
...	- further arguments for <code>parseKEGG_file()</code> .

### Details

The columns are automatically generated by the `parseKEGG_file` function into variables, which are further formatted specifically for the KEGG compound database.

The text file used is "KEGG\_path/ligand/compound/compound".

It decompresses "KEGG\_path/ligand/compound.tar.gz" if needed.

## Value

Generates `compound_reference_table` (.txt & .rda; saved to 'outDir') and returns a data frame with as many rows as entries and the following columns (or variables):

- (1) ID - C number identifier (e.g. "C00001");
- (2) NAME - compound name(s);
- (3) FORMULA - chemical formula;
- (4) EXACT\_MASS - compound's mass;
- (5) MOL\_WEIGHT - molecular weight;
- (6) REMARK - relationship with D number and others;
- (7) REACTION - reactions IDs (R#####) in which the compound is involved;
- (8) PATHWAY - pathway(s) in which the compound is involved (map### and name);
- (9) MODULE - module(s) in which the compound is involved (M##### and name);
- (10) ENZYME - EC numbers catalysing a reaction in which the compound is involved;
- (11) BRITE; (12) DBLINKS; (13) ATOM; (14) BOND; (15) COMMENT;
- (16) BRACKET; (17) SEQUENCE; (18) REFERENCE;

In all instances, multiple entries in a given column are separated by '['. EC numbers are of the form '\d[.]\\d+[.]\\d+[.]\\d+' (e.g. '1.97.1.12')

## See Also

`parseKEGG_file`

## Examples

```
KEGG_path <- "~/KEGG" # MODIFY TO KEGG PARENT FOLDER!
# The parent folder should contain the following (KEGG FTP structure):
# brite/
# genes/
# ligand/
# medicus/
# module/
# pathway/
# README.kegg
# RELEASE
# xml/

compound_reference_table <- parseKEGG_compound(KEGG_path)
# A .txt file (tab separated) is written to output/ (relative to current working directory)
```

---

`parseKEGG_enzyme`

*Parse the KEGG enzyme database*

---

## Description

Read the KEGG enzyme database text file and format it into a reference table.

## Usage

```
parseKEGG_enzyme(KEGG_path, outDir = "output", verbose = T, ...)
```

## Arguments

KEGG_path	- string pointing to the location of the KEGG database parent folder. The path to the required file is contained within the function.
outDir	- string pointing to the output folder. Default ("output/").
verbose	- logical. Should progress be printed to the screen? Default (TRUE)
...	- further arguments for parseKEGG_file().

## Details

The columns are automatically generated by the parseKEGG\_file function into variables, which are further formatted specifically for the KEGG enzyme database.

The text file used is "KEGG\_path/ligand/enzyme/enzyme".

It decompresses "KEGG\_path/ligand/enzyme.tar.gz" if needed.

## Value

Generates enzyme\_reference\_table (.txt & .rda; saved to 'outDir') and returns a data frame with as many rows as entries and the following columns (or variables):

- |                |  |
|----------------|--|
| (1) ID         | - Enzyme Commission (EC) number (e.g. "1.1.1.1"; 4 positions);   |
| (2) NAME       | - enzyme name(s);  |
| (3) CLASS_I    | - enzyme class; refers to the first position.<br>CLASSES:<br>1. Oxidoreductases, 2. Transferases, 3. Hydrolases,<br>4. Lyases, 5. Isomerases, 6. Ligases |
| (4) CLASS_II   | - further enzyme class info; refers to the second position<br>(different for every class);   |
| (5) CLASS_III  | - further enzyme class info; refers to the third position<br>(different for every class);  |
| (6) SYSNAME    | - alternative names;   |
| (7) REACTION   | - reaction(s) the enzyme catalyses;  |
| (8) ALL_REAC   | - reaction ID(s) (R number);   |
| (9) SUBSTRATE  | - substrate name and ID(s);  |
| (10) PRODUCT   | - product name and ID(s);  |
| (11) COMMENT;  | (12) HISTORY; (13) REFERENCE;  |
| (14) PATHWAY   | - pathway(s) in which the enzyme is involved (ec### and name);   |
| (15) ORTHOLOGY | - related KEGG Ortholog(s) (K number; K##### and name);  |
| (16) GENES;    | (17) DBLINKS;  |

\*In all instances, multiple entries in a given column are separated by '[';']'.

## See Also

parseKEGG\_file

## Examples

```
KEGG_path <- "~/KEGG" # MODIFY TO KEGG PARENT FOLDER!
# The parent folder should contain the following (KEGG FTP structure):
# brite/
# genes/
```



```

# ligand/
# medicus/
# module/
# pathway/
# README.kegg
# RELEASE
# xml/

enzyme_reference_table <- parseKEGG_enzyme(KEGG_path)
# A .txt file (tab separated) is written to output/ (relative to current working directory)

```

---

```

parseKEGG_execute_all Execute all parseKEGG parent functions to format KEGG databases
into data frames

```

---

## Description

Execute all parseKEGG parent functions to format specific KEGG databases into data frames.

## Usage

```
parseKEGG_execute_all(KEGG_path, ...)
```

## Arguments

KEGG_path	- string pointing to the location of the KEGG database parent folder.
...	- further arguments, such as outDir, for parseKEGG_file, parseKEGG_file.list and database-specific functions (below).

## See Also

parseKEGG\_compound, parseKEGG\_enzyme, parseKEGG\_genome, parseKEGG\_module,  
 parseKEGG\_ko, parseKEGG\_reaction, parseKEGG\_ko\_enzyme, parseKEGG\_ko\_reaction

## Examples

```

KEGG_path <- "~/KEGG" # MODIFY!
parseKEGG_parseKEGG_execute_all(KEGG_path)
# multiple reference_table objects in workspace and .txt files written to
#   output/ (relative to current working directory)

```

---

parseKEGG_file	<i>Parse any KEGG file without extension</i>
----------------	--

---

## Description

Read the KEGG database text file without extension (e.g. 'module', 'enzyme', 'genome') and format it into a reference table. Generates DATABASE\_reference\_table (data frame).

## Usage

```
parseKEGG_file(FILE_PATH, split_pattern = "ENTRY", pathway_trim = T,  
  verbose = T, ...)
```

## Arguments

split_pattern	- string to use to identify start of new section/entry. Default ("ENTRY").
pathway_trim	- logical. Should the file 'KEGG_path/pathway/pathway' be trimmed to only include 'map' entries? (i.e. exclude ko, ec and organism-specific pathways). Default (TRUE).
verbose	- logical. Should progress be printed to the screen? Default (TRUE).
...	- further arguments (currently unsupported)
FILE	- string pointing to the location of the file WITHOUT EXTENSION (uncompressed).

## Details

File must be decompressed before being processed. The functions that use this function (listed below in see also) perform this step if necessary.

## Value

A data frame with the formatted data. See the file-specific functions.

## See Also

parseKEGG\_compound, parseKEGG\_enzyme, parseKEGG\_genome, parseKEGG\_module,  
parseKEGG\_reaction, parseKEGG\_execute\_all

## Examples

```
compound_file_path <- "~/KEGG/ligand/compound/compound" # MODIFY!  
reference_table <- parseKEGG_file(compound_file_path)  
# WITHOUT DATABASE SPECIFIC FORMATTING!
```

---

`parseKEGG_file.list`      *Parse any '.list' KEGG file*

---

### Description

Reads the KEGG database text files with '.list' extension (e.g. 'ko\_enzyme.list', 'ko\_reaction.list') and formats it into a matrix with a binary indicator or relationships or mappings.

### Usage

```
parseKEGG_file.list(FILE_PATH, ...)
```

### Arguments

`FILE_PATH`      - string pointing to the location of '.list' file.  
...               - further arguments (currently unsupported)

### Value

MATRIX

### See Also

`parseKEGG_ko_enzyme`, `parseKEGG_ko_reaction`

### Examples

```
ko_enzyme_file_path <- "~/KEGG/genes/ko/ko_enzyme.list" # MODIFY!  
MAP <- parseKEGG_file(ko_enzyme_file_path)
```

---

`parseKEGG_genome`      *Parse the KEEG genome database*

---

### Description

Read and format the KEGG genome database text file and the organism information and format it into a reference table.

### Usage

```
parseKEGG_genome(KEGG_path, outDir = "output", addKOs = T, addECs = T,  
includeViruses = F, verbose = T, ...)
```

## Arguments

KEGG_path	- string pointing to the location of the KEGG database parent folder. The path to the required file(s) is contained within the function.
outDir	- optional. String pointing to the output folder. Default ("output/").
addKOs	- logical. Should the list of K numbers be retrieved for each organism? Default (TRUE).
addECs	- logical. Should the list of ECs be retrieved for each organism? Default (TRUE).
includeViruses	- logical. Should viral genomes be included? Default (FALSE).
verbose	- logical. Should progress be printed to the screen? Default (TRUE).
...	- further arguments for parseKEGG_file().

## Details

The columns are automatically generated by the parseKEGG\_file function into variables, which are further formatted specifically for the KEGG genome database.

The main text file used is "KEGG\_path/genes/genome/genome".

It decompresses "KEGG\_path/genes/genome.tar.gz" if needed.

If addECs and/or addKOs are set to TRUE, the KEGG genome 3-4 letter code identifier are used to retrieve the enzyme and KEGG ortholog content for each KEGG genome, respectively.

## Value

Generates genome\_reference\_table (.txt & .rda; saved to 'outDir') and returns a data frame with as many rows as entries and the following columns (or variables):

(1) ID	- KEGG genome identifier (T0 number; e.g. "T00001");
(2) ORG_ID	- KEGG organism identifier (3 or 4 letter code; e.g. "hin");
(3) STATUS	- genome sequence status (e.g. "Complete Genome");
(4) NAME	- various identifiers (e.g. "hin, HAEIN, 71421");
(5) ORGANISM	- organism name (Genus species sp);
(6) ANNOTATION	- type of annotation (one of "manual", "KOALA" or "none");
(7) TAXONOMY;	(8) DATA_SOURCE;
(9) ORIGINAL_DB	- original database;
(10) KEYWORDS;	(11) DISEASE;
(12) COMMENT;	(13) CHROMOSOME;
(14) STATS_N_NUCLEOTIDES	- statistics, number of nucleotides;
(15) STATS_N_GENES_PROT	- statistics, number of protein-encoding genes;
(16) STATS_N_GENES_RNA	- statistics, number of RNA-encoding genes;
(17) REFERENCE	- reference(s) for study from which genomic sequence was derived;
(18) PLASMID;	(19) DBLINKS;
(20) KOs	- concatenated string of KEGG Orthologs (K numbers; e.g. "K00001");
(21) ECs	- concatenated string of Enzyme Classification (EC) numbers (e.g. "1.1.1.1").

\*In all instances, multiple entries in a given column are separated by '[';']'.

## See Also

parseKEGG\_file

## Examples

```
KEGG_path <- "~/KEGG" # MODIFY TO KEGG PARENT FOLDER!
# The parent folder should contain the following (KEGG FTP structure):
# brite/
# genes/
# ligand/
# medicus/
# module/
# pathway/
# README.kegg
# RELEASE
# xml/

genome_reference_table <- parseKEGG_genome(KEGG_path)
# A .txt file (tab separated) is written to output/ (relative to current working directory)
```

---

parseKEGG_ko	<i>Parse the KEGG orthology (KO) database</i>
--------------	---

---

## Description

Read and format the KEGG orthology (KO) database (containing ortholog (gene) information) text file into a reference table.

## Usage

```
parseKEGG_ko(KEGG_path, outDir = "output", verbose = T, ...)
```

## Arguments

KEGG_path	- string pointing to the location of the KEGG database parent folder. The path to the required file is contained within the function.
outDir	- optional. String pointing to the output folder. Default ("output/").
verbose	- logical. Should progress be printed to the screen? Default (TRUE).
...	- further arguments for parseKEGG_file().

## Details

The columns are automatically generated by the parseKEGG\_file function into variables, which are further formatted specifically for the KEGG ortholog database.

The text file used is "KEGG\_path/genes/ko/ko".

It decompresses "KEGG\_path/genes/ko.tar.gz" if needed.

## Value

Generates ko\_reference\_table (.txt & .rda; saved to 'outDir') and returns a data frame with as many rows as entries and the following columns (or variables):

\*In all instances, multiple entries in a given column are separated by '[';']'.

**See Also**

parseKEGG\_file

**Examples**

```
KEGG_path <- "~/KEGG" # MODIFY TO KEGG PARENT FOLDER!
# The parent folder should contain the following (KEGG FTP structure):
# brite/
# genes/
# ligand/
# medicus/
# module/
# pathway/
# README.kegg
# RELEASE
# xml/

ko_reference_table <- parseKEGG_ko(KEGG_path)
# A .txt file (tab separated) is written to output/ (relative to current working directory)
```

---

parseKEGG_ko_enzyme	<i>Map KEGG orthology (KO) to Enzyme Commission (EC) numbers</i>
---------------------	--

---

**Description**

Map KEGG orthologs (K numbers) to EC numbers and format it into a matrix with binary indicator for mapping/relationship. Generates ko\_enzyme\_map (.txt & .rda)

**Usage**

```
parseKEGG_ko_enzyme(KEGG_path, outDir = "output", verbose = T, ...)
```

**Arguments**

- KEGG\_path - string pointing to the location of the KEGG database parent folder. The path to the required file is contained within the function.
- outDir - string pointing to the output folder. Default ("output/").
- verbose - logical. Should progress be printed to the screen? Default (TRUE)
- ... - further arguments for parseKEGG\_file.list().

**Value**

Data frame establishing the relationship between K numbers and enzymes (binary).

```
> ko_enzyme_map[1:3,1:3]

      1.1.1.1 1.1.1.10 1.1.1.100
K00001      1        0        0
K00002      0        0        0
K00003      0        0        0
```

## See Also

parseKEGG\_file.list

## Examples

```
KEGG_path    <- "~/KEGG" # MODIFY TO KEGG PARENT FOLDER!
# The parent folder should contain the following (KEGG FTP structure):
# brite/
# genes/
# ligand/
# medicus/
# module/
# pathway/
# README.kegg
# RELEASE
# xml/

ko_enzyme_map <- parseKEGG_ko_enzyme(KEGG_path)
# A .txt file (tab separated) is written to output/ (relative to current working directory)
```

---

parseKEGG\_ko\_reaction *Map KEGG orthologs (KOs) to Reaction IDs*

---

## Description

Map KEGG orthologs (KOs) to Reaction IDs and format it into a matrix with binary indicator for mapping/relationship. Generates 'ko\_reaction\_map' (.txt & .rda).

## Usage

```
parseKEGG_ko_reaction(KEGG_path, outDir = "output", verbose = T, ...)
```

## Arguments

KEGG_path	- string pointing to the location of the KEGG database parent folder. The path to the required file is contained within the function.
outDir	- string pointing to the output folder. Default ("output/").
verbose	- logical. Should progress be printed to the screen? Default (TRUE).
...	- further arguments for parseKEGG_file.list().

## Value

Data frame establishing the relationship between KO numbers and reactions (R number) (binary).

```
> ko_reaction_map[1:3,1:3]

      R00005 R00006 R00008
K00001      0      0      0
K00002      0      0      0
K00003      0      0      0
```

## See Also

parseKEGG\_file.list

## Examples

```
KEGG_path      <- "~/KEGG" # MODIFY TO KEGG PARENT FOLDER!
# The parent folder should contain the following (KEGG FTP structure):
# brite/
# genes/
# ligand/
# medicus/
# module/
# pathway/
# README.kegg
# RELEASE
# xml/

ko_reaction_map <- parseKEGG_ko_reaction(KEGG_path)
# A .txt file (tab separated) is written to output/ (relative to current working directory)
```

---

parseKEGG_module	<i>Parse the KEGG module database</i>
------------------	---------------------------------------

---

## Description

Read the KEGG module database text file and format it into a reference table.

## Usage

```
parseKEGG_module(KEGG_path, outDir = "output", verbose = T,
  shortName_file_path = "", ...)
```

## Arguments

KEGG_path	- string pointing to the location of the KEGG database parent folder. The path to the required file is contained within the function.
outDir	- string pointing to the output folder. Default ("output/"). NULL overwrites existing files.
verbose	- logical. Should progress be printed to the screen? Default (TRUE)
shortName_file_path	- file path to table containing Module IDs (column 1), Short name (column 2). Default ("")
...	- further arguments for parseKEGG_file()



## Details

The columns are automatically generated by the `parseKEGG_file` function into variables, which are further formatted specifically for the KEGG module database.

The text file used is "KEGG\_path/module/module".

It decompresses "KEGG\_path/module/module.gz" if needed.

DEFINITION: Logical Expression (adapted from <http://www.genome.jp/kegg/module.html>)

The MODULEs (identified by an M number; e.g. "M00001") are defined by a logical expression (DEFINITION) of KEGG orthologs (KO numbers, KOs; e.g. "K00001") and sometimes other M numbers, facilitating the automatic evaluation of whether a module is complete in a given genome.

A MODULE is made up of BLOCKS (SPACE delimited). Each block is defined by a logical expression to determine which KOs are needed in the definition. All BLOCKS must be present to be able to state that a MODULE is COMPLETE.

The KEGG module DEFINITION has been formatted to simplify its use, but the logical expression is conserved. Where space or a plus sign represent AND operations in the KEGG definition, we have replaced all instances with '&'. Similarly, we have replaced all comas (used within KEGG to represent an OR operation) with pipes ('|').

We have also translated the K number based DEFINITION to an enzyme based DEFINITION using the ORTHOLOGY information. K numbers can be mapped to enzymes using the Enzyme Classification (EC) numbers (redundancy expected). Note that not all K numbers have an association to an EC number. In these instances, the K numbers have been left in the MODULE DEFINITION. When evaluating whether a MODULE is complete using EC numbers, the user must decide whether to assume that those genes are present or not (default). See `query_genomes_to_modules`

Optional items in the complex or definition (denoted by a minus sign) have been removed from the main definition and listed as a separate column under 'OPTIONAL\_' (EC or K numbers).

`shortName_file_path` can be used to provide the path to a file listing:

```
Column named 'ID'           - KEGG module ID (M numbers),
Column named 'NAME'         - KEGG module NAME,
Column names 'NAME_SHORT'   - manually abbreviated names for plotting purposes.
```

NAME\_SHORT abbreviations: 2-CRS, two-component regulatory system; PS, photosystem/photosynthesis; pwy, pathway; TS, transport system; R, resistance;

## Value

Generates `module_reference_table` (.txt & .rda; saved to 'outDir') and returns a data frame with as many rows as entries and the following columns (or variables):

- (1) ID - KEGG module identifier (M number; e.g. "M00001");
- (2) NAME - KEGG module name;
- (3) NAME\_SHORT - abbreviated module name (for visualization purposes);
- (4) DEFINITION\_KOs - module definition as a logical expression (see Details) in terms of KEGG Orthologs;
- (5) DEFINITION\_EC - module definition as a logical expression (see Details) in terms of EC numbers;
- (6) OPTIONAL\_KOs - optional K numbers listed;
- (7) OPTIONAL\_EC - optional EC numbers listed;
- (8) ORTHOLOGY - relationship between K and EC numbers;
- (9 - 11) CLASS\_I - III - hierarchical module classes;

(12) PATHWAY	- pathway(s) in which the module is involved (map### and name);
(13) REACTION	- reaction(s) in which the module is involved (R##### and their corresponding compound IDs);
(14) COMPOUND	- compound(s) in which the module is involved (C#####);
(15) DBLINKS;	(16) RMODULE; (17) REFERENCE;
(18) COMMENT;	(19) BRITE;

\*In all instances, multiple entries in a given column are separated by '[';']'.

## See Also

```
parseKEGG_file, misc_module_definition_check,
misc_module_definition_optional,' misc_module_definition_block_EC
```

## Examples

```
KEGG_path <- "~/KEGG" # MODIFY TO KEGG PARENT FOLDER!
# The parent folder should contain the following (KEGG FTP structure):
# brite/
# genes/
# ligand/
# medicus/
# module/
# pathway/
# README.kegg
# RELEASE
# xml/

module_reference_table <- parseKEGG_module(KEGG_path)
# And .txt file written to output/ (relative to current working directory)
```

---

```
parseKEGG_process_KEGG_taxonomy
```

*Retrieve the taxonomy listed as part of the KEGG genome database.*

---

## Description

Retrieve the taxonomy listed as part of the KEGG genome database.

## Usage

```
parseKEGG_process_KEGG_taxonomy(genome_reference_table,
  taxonomy_header = "TAXONOMY", org_header = 1, ...)
```

## Arguments

```
genome_reference_table
- table containing the different genome entries (rows) and data (columns). See
  parseKEGG_genome.
```

taxonomy_header	- optional. Character with header name or number indicating column index for taxonomic information. Default ("TAXONOMY").
org_header	- optional. Character with header name or number indicating column index for organism name or ID. Use to name the output rownames. Default (1; first column).
...	- further arguments (currently unsupported)

## Details

After processing the KEGG genome database, this function can extract the taxonomic information. This is obtained from splitting the TAXONOMY column after the "LINEAGE" tag into "words". This information should specify KINGDOM, PHYLUM, CLASS, ORDER, FAMILY, GENUS (6), but there are occasionally more or less words than expected. Therefore NOTE that it is incomplete and inconsistent, most likely because it is derived from multiple sources. Use with caution.

## Value

Data frame with 6 columns containing the taxonomic information (columns: KINGDOM, PHYLUM, CLASS, ORDER, FAMILY, GENUS). The rownames contain the organism information (name or ID) as specified with org\_header. UNKNOWN label is added where no information is available.

## See Also

parseKEGG\_genome

## Examples

```
# Generate KEGG's genome database table
genome_reference_table <- parseKEGG_genome("~/KEGG")

# Extract taxonomic information from genome_referene_table
TAXONOMY_table <- parseKEGG_process_KEGG_taxonomy(genome_reference_table)
TAXONOMY_table[1,]
#      KINGDOM  PHYLUM      CLASS      ORDER      FAMILY      GENUS
# T00001  Bacteria  Proteobacteria  Gammaproteobacteria  Pasteurellales  Pasteurellaceae  Haemophilus
```

---

parseKEGG_reaction	<i>Parse KEGG reaction database</i>
--------------------	-------------------------------------

---

## Description

Read and format the KEGG reaction database text file into a reference table.

## Usage

```
parseKEGG_reaction(KEGG_path, outDir = "output", verbose = T, ...)
```

## Arguments

KEGG_path	- string pointing to the location of the KEGG database parent folder. The path to the required file is contained within the function.
outDir	- string pointing to the output folder. Default ("output/").
verbose	- logical. Should progress be printed to the screen? Default (TRUE)
...	- further arguments for parseKEGG_file().

## Details

The columns are automatically generated by the parseKEGG\_file function into variables, which are further formatted specifically for the KEGG reaction database.

The text file used is "KEGG\_path/ligand/reaction/reaction".

It decompresses "KEGG\_path/ligand/reaction.tar.gz" if needed.

## Value

Generates reaction\_reference\_table (.txt & .rda; saved to 'outDir') and returns a data frame with as many rows as entries and the following columns (or variables):

- (1) ID - R number identifier (e.g. "R00001");
- (2) NAME - reaction or enzyme name;
- (3) DEFINITION - reaction definition using compound's names;
- (4) EQUATION - reaction definition using compound's IDs;
- (5) ENZYME - Enzyme Commission (EC) number (e.g. "1.1.1.1");
- (6) COMMENT; (7) RCLASS;

In all instances, multiple entries in a given column are separated by '[';']'.

## See Also

parseKEGG\_file

## Examples

```
KEGG_path <- "~/KEGG" # MODIFY TO KEGG PARENT FOLDER!
# The parent folder should contain the following (KEGG FTP structure):
# brite/
# genes/
# ligand/
# medicus/
# module/
# pathway/
# README.kegg
# RELEASE
# xml/

reaction_reference_table <- parseKEGG_reaction(KEGG_path)
# A .txt file (tab separated) is written to output/ (relative to current working directory)
```

---

plot_heatmap	<i>Plot a heatmap</i>
--------------	-----------------------

---

## Description

Plot a heatmap of the the module fraction present across genomes or of the variance across groups.

## Usage

```
plot_heatmap(data_in, row_lab = "Genomes", col_lab = "Modules",
  ORDER_MATRIX = TRUE, legend_name = "Module\ncompleteness\nfraction\n",
  set_yLim = FALSE, Filename = "", Width = 24, Height = 18, ...)
```

## Arguments

data_in	- numeric matrix OR 3-column data frame. See Details.
row_lab	- optional. String to specify the axis label corresponding to the row values. Default ("Genomes").
col_lab	- optional. String to specify the axis label corresponding to the column values. Default ("Modules").
ORDER_MATRIX	- logical. Should rows and columns be reorder according to the dendrogram (hierarchical clustering). Default (TRUE).
legend_name	- optional. String to specify the legend title. Default ("Fraction\n matched\n").
set_yLim	- optional. Logical or numeric (length 2) indicating whether to set the y limit of the heatmap scale. Default (FALSE). See Details.
Filename	- optional. A character vector containing the file path to save image(s). The device to save is determined by file extension. Default ("", i.e. file not written).
Width	- optional. Width size for file. Default (24 in).
Height	- optional. Height size for file. Default (18 in).
...	- further arguments (currently unsupported)

## Details

If data\_in is a data frame, the heatmap will be made using the first column as the row entry labels, the second column as the column entry labels and the third as the actual value to be plotted.

If data\_in is a numeric matrix, it will be 'melted' into this type of data frame by using the reshape2::melt function.

set\_yLim - #' the default (FALSE), sets the y limit to c(0, 1), while TRUE to the c(min, max) of the data. If numeric, the values given are used for the lower and upper limits respectively.

## Value

ggplot object. Saves figures if Filename is specified.

## Examples

```
data(data_example_moduleIDs)
data(data_example_genomeIDs)

# Calculate the module completion fraction (mcf) for the genomes
# and modules contained in the data objects above.
OUT <- query_genomes_to_modules(data_example_genomeIDs,MODULE_ID = data_example_moduleIDs)

# Make a heatmap of the mcf output from query_genomes_to_modules
# Rows and columns are reordered according to the dendrogram resulting from
# hierarchical clustering by default.
p <- plot_heatmap(OUT$MATRIX,Filename = "plot_heatmap.png")
```

---

plot_scatter	<i>Scatter plot</i>
--------------	---------------------

---

## Description

Scatter plot for pairs of categorical data with a numeric value.

## Usage

```
plot_scatter(plot_data, X = 1, Y = 3, colBy = NULL, xLab = "",
  yLab = "", xLabs_angle = T, Filename = "plot_scatter.pdf", Width = 24,
  Height = 18, ...)
```

## Arguments

plot_data	- data frame. The columns to be plotted are indicated with X and Y. Order given is conserved.
X, Y	- optional. Number indicating column to use for x axis and y axis, respectively. Default (1 and 3).
colBy	- optional. Number indicating column to use for coloring grouping. Default (NULL).
xLab, yLab	- optional. String to use for x label and y label, respectively. Default (first and second column names, respectively).
xLabs_angle	- optional. Should x-axis labels be rotated 45 degrees? Default (TRUE).
Filename	- optional. String(s) to use for file name. Default ("plot_scatter.pdf"). If set to "" a file is not written.
Width	- width for figure file. Default (24in).
Height	- height for figure file. Default (18in).
...	- further arguments (currently unsupported)

## Details

This function is used by `analysis_genomes_module_output()` to plot the module variance accross genomes and

by `analysis_pca_mean_distance_grouping()` to plot the mean PCA distance of the groups analysed.

If `colBy` is set to `NULL`, no grouping will be done for colouring (a scale warning will be issued that can be safely ignored).

## Value

ggplot2 plot object

## See Also

analysis\_genomes\_module\_output, analysis\_pca\_mean\_distance\_grouping

## Examples

```
plot_data_example <- data.frame("Groups"=LETTERS[1:12],
                                "Factor"=c(rep(1,4),rep(2,4),rep(3,4)),
                                "Value"=runif(12,-10,10),stringsAsFactors = FALSE)

plot_scatter(plot_data_example,Filename = "")

# Change plot order to be according to the numeric value
plot_data_example <- plot_data_example[order(plot_data_example$Value),]
plot_scatter(plot_data_example,Filename = "")
```

---

plot\_scatter\_byFactors

*Scatter plot with overlapping factors/groups.*

---

## Description

Represent different groups as defined by FACTOR(S) in a scatter plot.

## Usage

```
plot_scatter_byFactors(MATRIX, FACTOR, factor_labs = NULL, xLab = NULL,
  yLab = NULL, Filename = "plot_scatter_byFactors.pdf", Width = 7,
  Height = 5, ...)
```

## Arguments

MATRIX	- two column matrix to plot. Only the first two columns will be used.
FACTOR	- character vector or list of character vectors used to split the data into groups.
factor_labs	- optional. Character vector to distinguish FACTOR groups. Default (NULL). See Details.
Filename	- optional. Character vector containing the file path to save the image. Must have an extension (see Details). Default ("plot_scatter_byFactors.pdf"). If set to "", no file will be written.
Width	- Width size for file. Default (7 in).
Height	- Height size for file. Default (5 in).
...	- further arguments (currently unsupported)

## Details

This function is used by `analysis_genomes_module_output()` to plot the first two Principal Components (PCs) from the PCA analysis, overlapping different factors or groups (as determined by FACTOR). It is also used by `analysis_pca_mean_distance_grouping()` to highlight a single group on PC plot.

`factor_labs` is used as an extension for the filename for the plot files. The names of the FACTOR object OR "factor" followed by a number will be used if `factor_labs` is not specified (i.e. `factor_labs = NULL`).

A plot is only generated for FACTORS with less than 60 groups. All the data is plotted in grey in the background, with groups being overlayed for each Factor.

Only the following file types are allowed: pdf, png, svg and jpeg.

## Value

A list with as many entires as factors is generated (one for each factor) using `factor_labs` for the names. For every FACTOR, a list will contain:

```
$nGroups - numeric. The number of groups found for that FACTOR
              NOTE: That entries with "" will be excluded.
$table    - data frame with the number of entries for each group found.
$file     - character vector of file name(s).
```

## See Also

`analysis_genomes_module_output`, `analysis_pca_mean_distance_grouping`

## Examples

```
data(data_example_moduleIDs)
data(data_example_genomeIDs)
# length(data_example_genomeIDs) # [1] 25

# Calculate the module completion fraction (mcf) for the genomes
# and modules contained in the data objects above.
OUT <- query_genomes_to_modules(data_example_genomeIDs,
                                MODULE_ID = data_example_moduleIDs)

pca <- prcomp(OUT$MATRIX)

# Make boxplots of the mcf output from query_genomes_to_modules
this_FACTOR <- rep(LETTERS[1:5],length(data_example_genomeIDs)/5)
plot_output <- plot_scatter_byFactors(pca$x[,1:2],FACTOR = this_FACTOR,
                                     factor_labs = "random",
                                     Filename = "plot_scatter_byFactors.png")

# NAs are ommitted, so a single group can be contrasted with overall data
this_FACTOR <- c(rep(NA,20),rep(LETTERS[1],5))
plot_output <- plot_scatter_byFactors(pca$x[,1:2],FACTOR = this_FACTOR,
                                     factor_labs = "group_A",
                                     Filename = "plot_scatter_byFactors_single.png")
```



---

plot_sunburst	<i>Sunburst plot</i>
---------------	----------------------

---

## Description

This function arranges the hierarchical data and fills based on the counts for each category and level.

## Usage

```
plot_sunburst(DATA, centerLabel = "", ANGLE = FALSE, fill_by = NULL,
  fill_by_mean = FALSE, outer_text = TRUE,
  outer_text_levelN_minus_1 = TRUE, legend_name = "Counts\n",
  sunburst = TRUE, setMax = NULL, setMin = NULL, textSize = NULL,
  textColour = "black", Filename = "", WIDTH = 25, HEIGHT = 25, ...)
```

## Arguments

DATA	- Data frame containing hierarchical data by columns, where the left column is the highest level and the right column the lowest one.
centerLabel	- optional. String to be used for centre label. Default (""; i.e. no text).
ANGLE	- optional. Should the angle of the text be adjusted to the position where it's at? Default (FALSE; i.e. horizontal text).
fill_by	- optional. Numeric vector containing values to be used to fill outer most ring (see Details). Default (NULL).
fill_by_mean	- optional. Should the categories in the inner rings be filled (colored) by the mean of the fill_by value provided. Default (FALSE).
outer_text	- optional. Should the text from the lowest level (right column) be included? Default (TRUE).
outer_text_levelN_minus_1	- optional. Should the text from the second lowest level (second column from right) be included? Default (TRUE).
legend_name	- optional. Default ("Counts\n"; i.e. have space between name and legend bar).
sunburst	- optional. Should the sunburst be made (default) or should the output be bars? Default (TRUE, ).
setMax	- optional. Numerical value to set the maximum limit. Categories with higher values than those specified will be excluded, making them light blue. Default (NULL).
setMin	- optional. Numerical value to set the minimum limit. Categories with lower values than those specified will be excluded, making them light blue. Default (NULL).
textSize	- optional. Numerical value to set the text size. Default (NULL).
textColour	- optional. String to set the text colour. Default ("black").
Filename	- optional. A character vector containing the file path to save image(s). The device to save is determined by file extension. Default (""; i.e. printed to device, file not written).
WIDTH	- optional. Width size for file. Note that text size in plot scales with WIDTH. Default (25 in).
HEIGHT	- optional. Height size for file. Default (25 in).

## Details

This function arranges a hierarchical data set into a dart-style chart, which is then partitioned and colored based on the data structure. The outer most section displays the number of elements (counts) in the lowest (most specific) hierarchical level. The user has the option to fill the outer level with user specified values given to `fill_by` or to fill with the mean value `fill_by_mean`. See Details.

The sunburst is generated by having a ring containing the hierarchical data from highest level (most generic) at the centre, followed by concentric rings that go out as the data becomes more specific. The outer-most ring is colored (filled) by the number of members (counts) in that category. The hierarchy is conserved throughout the data levels.

LEVEL_1	LEVEL_2	LEVEL_3
a	aa	aa1
a	aa	aa2
a	ab	ab1
a	ab	ab2
b	ba	ba1
b	ba	ba2
b	bb	bb1
b	bb	bb2
c	ca	ca1
c	ca	ca2
c	cb	cb1
c	cb	cb2

The above data would result in a sunburst plot that would have 3 categories in the inner-most ring, each with 2 categories in the second ring (6 total) and 12 categories in the outer ring, one for each of the second level categories. Therefore, the counts would all be 1 (see the first example for a case with different levels).

`fill_by` allows the user to display other values associated with the outer-most categories, such as the module fraction completeness (`mfc`), rather than the membership count. If `fill_by_mean` is set to `TRUE`, then the inner rings are also filled (colored) based on the mean of the `fill_by` value of the data making up the category.

`fill_by` should contain as many entries as there are categories at the last level. If more are provided, only the required entries will be used (i.e. if 5 values are provided for data with 3 categories, only the first 3 values will be used). Only as many entries as there are rows in `DATA` are allowed.

Function developed by expanding and modifying Yahia El Gamal's blog post "Create Basic Sunburst Graphs with ggplot2" (<https://medium.com/optima-blog/create-basic-sunburst-graphs-with-ggplot2-7d7484d92c61>)

## Value

List containing the `$DATA` used to generate the plot and the plot itself as a ggplot object `$p`.

## Examples

```
data(data_example_sunburst)
```

#	CLASS_I	CLASS_II	CLASS_III	NAME_SHORT
#	Pathway module	Carbohydrate & lipid metabolism	Fatty acid metabolism	beta-Oxidation
#	Pathway module	Energy metabolism	Methane metabolism	Methanogenesis, from methanol
#	Pathway module	Energy metabolism	Methane metabolism	Methanogenesis, from acetate
#	Pathway module	Energy metabolism	Carbon fixation	Reductive acetyl-CoA pathway

```

# Pathway module      Energy metabolism  Nitrogen metabolism  Dissimilatory nitrate reduction
# Pathway module      Energy metabolism  Methane metabolism   Methanogenesis, from methylamine
# Pathway module      Energy metabolism  Methane metabolism   Methanogenesis, from CO2
# Pathway module      Energy metabolism  Sulfur metabolism    Dissimilatory sulfate reduction

# Simplest plot using count data.
plot_data <- plot_sunburst(sunburst_data, WIDTH = 8) # WIDTH scales text size

# Specify values to be used for outer ring (lowest level) and change legend name accordingly

data(data_example_sunburst_fill_by)
# mcf for the modules associated with the data shown above for an example dataset.
plot_data <- plot_sunburst(data_example_sunburst, centerLabel = "Org A",
                           fill_by = data_example_sunburst_fill_by, outer_text = F, WIDTH = 8,
                           legend_name = "fill_by")

# Also fill inner rings (levels) according to the mean values determined by fill_by.
plot_data <- plot_sunburst(data_example_sunburst, centerLabel = "Org A",
                           fill_by = data_example_sunburst_fill_by, fill_by_mean = T,
                           outer_text = F, WIDTH = 8, legend_name = "fill_by")

```

---

`plot_variance_boxplot` *Make a boxplot*

---

## Description

Make a boxplot

## Usage

```

plot_variance_boxplot(MATRIX_IN, x_lab = "Modules",
  y_lab = "Fraction matched", xLabs_angle = T, Filename = "",
  Width = 24, Height = 18, ...)

```

## Arguments

<code>MATRIX_IN</code>	- numeric matrix containing the module fraction match of the genomes (rows) and modules (columns). The columns are used as factor for the box plot grouping.
<code>x_lab</code>	- optional. String to specify the x-axis label corresponding to the row values. Default ("Modules").
<code>y_lab</code>	- optional. String to specify the legend title. Default ("Fraction\n matched").
<code>xLabs_angle</code>	- optional. Should x-axis labels be rotated 45 degrees? Default (TRUE).
<code>Filename</code>	- optional. A character vector containing the file path to save image(s). The device to save is determined by file extension. Default ("", i.e. file not written).
<code>Width</code>	- Width size for file. Default (24in).
<code>Height</code>	- Height size for file. Default (18in).
<code>...</code>	- further arguments (currently unsupported)

## Value

ggplot object of plot

---

query\_genes\_to\_genomes

*Find the genome(s) that contain a set of genes.*

---

## Description

The genes can be either an enzyme, given by its Enzyme Classification (EC) number (e.g. "1.1.1.1"), or a KEGG ortholog identifier (K number, e.g. "K00001").

## Usage

```
query_genes_to_genomes(genes, use_genome_reference_table = NULL,  
  use_ko_reference_table = NULL, use_enzyme_reference_table = NULL, ...)
```

## Arguments

genes	- character vector of KO or EC numbers. See Details for more info on the use of ECs.
use_genome_reference_table	- optional. Provide a data frame with updated KEGG genome database. Default (NULL; inbuilt data used). See Details.
use_ko_reference_table	- optional. Provide a data frame with updated KEGG ortholog database. Default (NULL; inbuilt data used). See Details.
use_enzyme_reference_table	- optional. Provide a data frame with updated KEGG enzyme database. Default (NULL; inbuilt data used). See Details.
...	- further arguments (currently unsupported)

## Details

When providing EC numbers, the user can provide a full EC number as described above or the first three values/positions (e.g. "1.1.1" and "1.1.1.-" are both allowed). In the latter case, the function evaluates all enzymes (EC numbers) that match the three values/positions provided (e.g., using "1.1.1" would cause the function to evaluate all enzymes starting with that EC number combination). In other words, using a three number entry would act as a wildcard functioning for the 4th position of the EC notation.

The use\_ set of arguments allow users with KEGG FTP access to provide the updated data from the KEGG databases in the form of reference tables AND/OR for advanced users to provide custom-made modules (see below). These reference tables can be generated with the parseKEGG family of functions and need to have a specific format (see function descriptions for details on format).

If providing use\_genome\_reference\_table, make sure that the parseKEGG\_genome function is run with arguments addECs = T and/or addKOs = T to include genes that make up the genomes.

ko\_reference\_table and enzyme\_reference\_table are used to verify that the genes provided exist within the KEGG database.

## Value

Data frame containing the genes (rows, specified in the rownames) and the T number of the KEGG genomes (columns) with a binary indicator for presence of gene in given genome.

## See Also

`parseKEGG_genome`, `ko_reference_table`, `enzyme_reference_table`

## Examples

```
genomes <- query_genes_to_genomes("K00844")

genomes <- query_genes_to_genomes("1.1.1.1")

genomes <- query_genes_to_genomes(genes = paste("K0000", 1:3, sep=""))
```

---

`query_genes_to_modules`

*Given a set of genes, find the modules they are involved in.*

---

## Description

The genes can be either enzymes, given by its Enzyme Classification (EC) number (e.g. "1.1.1.1"), or KEGG ortholog identifiers (K number, e.g. "K00001"). Using EC or K numbers might give different results, as an EC number might map to multiple K numbers or none.

## Usage

```
query_genes_to_modules(genes, use_module_reference_table = NULL,
  use_ko_reference_table = NULL, use_enzyme_reference_table = NULL, ...)
```

## Arguments

<code>genes</code>	- character vector (length 1). K or EC number.
<code>use_module_reference_table</code>	- optional. Provide a data frame with updated KEGG module database OR with custom-made modules. Default (NULL; inbuilt data used). See Details.
<code>use_ko_reference_table</code>	- optional. Provide a data frame with updated KEGG ortholog database. Default (NULL; inbuilt data used). See Details.
<code>use_enzyme_reference_table</code>	- optional. Provide a data frame with updated KEGG enzyme database. Default (NULL; inbuilt data used). See Details.
<code>...</code>	- further arguments (currently unsupported)

## Details

The `use_` set of arguments allow users with KEGG FTP access to provide the updated data from the KEGG databases in the form of reference tables AND/OR for advanced users to provide custom-made modules (see below). These reference tables can be generated with the `parseKEGG` family of functions and need to have a specific format (see function descriptions for details on format).

The module definition (contained in `module_reference_table`) describes the relationship between genes and modules and is used to identify the modules in which genes is involved. The user can

provide custom-made module definitions that use the logical expression format (however, the table format must be conserved!).

If enzyme identifiers are provided, note that as there are lingering unspecified enzymes (e.g. '1.1.1.-') in the KEGG module EC-based definition, this function also returns a row entry for the unspecified-versions of enzymes provided involved in one or more modules.

## Value

Data frame containing a binary indicator for genes (rows, specified by rownames) involved in modules (columns, Module IDs specified in names). A column called 'no\_match' is returned if one or more of the genes is not involved in any modules.

NULL is returned when there are no valid entries to evaluate. Note that enzyme entries are checked for 4 position completeness.

## See Also

`parseKEGG_module`

## Examples

```
modules <- query_genes_to_modules("K00844")

modules <- query_genes_to_modules("1.1.1.1")
```

---

```
query_genomes_to_modules
```

*Evaluates the KEGG modules presence given genome information.*

---

## Description

This function returns the 'completeness' of KEGG modules in the provided genome(s) (either as a genome identifier or as a lists of Enzyme Commission (EC) number or KEGG ortholog identifier, K number). The user can define which modules the function should check by providing a single or set of modules under 'MODULE\_ID'. If this is left blank, the function returns completeness of all KEGG modules (excluding modules defined in terms of other modules). The function can also be restricted to a subset of all KEGG modules by using the SEARCH\_NAME and SEARCH\_CLASS arguments. See Details.

## Usage

```
query_genomes_to_modules(GENOME_INFO, splitBy = "[;]", GENOME_ID_COL = 1,
  GENES_COL = 2, MODULE_ID = "", SEARCH_NAME = "", SEARCH_CLASS_I = "",
  SEARCH_CLASS_II = "", SEARCH_CLASS_III = "", EXCLUDE_NAME = "",
  OUT_MODULE_NAME = FALSE, META_OUT = FALSE, ADD_OUT = FALSE,
  use_genome_reference_table = NULL, ...)
```

## Arguments

GENOME_INFO	- character vector containing genome identifier(s) or organism name(s) OR data frame containing genome identifier/names(s) and gene list. See Details.
splitBy	- string indicating the split pattern for the data contained in the column indicated by GENES_COL. Default ("[:]" : uses ' ; ' , the ' [ ] ' indicates that it is NOT a regular expression).
GENOME_ID_COL	- optional. Column NAME or NUMBER containing genome NAME or IDENTIFIER. Default (1; <first column>). See Details.
GENES_COL	- optional. Column NAME or NUMBER pointing to the GENES. Default (2; <second column>). See Details.
MODULE_ID	- optional. Character vector listing specific KEGG module IDs (e.g. "M00001"). Default ("").
SEARCH_NAME	- optional. Character vector listing terms to search in KEGG module NAME field (case-insensitive). Default ("").
SEARCH_CLASS_I	- optional. Character vector listing terms to search in KEGG module CLASS_I field (case-insensitive). Default ("").
SEARCH_CLASS_II	- optional. Character vector listing terms to search in KEGG module CLASS_II field (case-insensitive). Default ("").
SEARCH_CLASS_III	- optional. Character vector listing terms to search in KEGG module CLASS_III field (case-insensitive). Default ("").
EXCLUDE_NAME	- optional. Character vector listing terms that if matched in KEGG module NAME field will be excluded (case-insensitive). Default ("").
OUT_MODULE_NAME	- optional (logical). Should the column names of MATRIX be the module IDs (M numbers) or module names? Default (FALSE; return matrix with M numbers)
META_OUT	- optional (logical). Should the KEGG module metadata be outputted? Default (FALSE).
ADD_OUT	- optional (logical). Should additional information be outputted? Default (FALSE).
use_genome_reference_table	- optional. Provide a data frame with updated KEGG module database. Default (NULL; inbuilt data used). See Details.
...	- further arguments passed to misc_geneVector_module, such as KO_in_DEF_EC or use_module_reference_table. See Details.

## Details

This function processes the GENOME\_INFO by passing each in turn to misc\_geneVector\_module() and collating all the output. GENOME\_ID\_COL and GENES\_COL are only used if GENOME\_INFO is a data frame. Post-analysis of this output can be carried out with analysis\_genomes\_module\_output().

Appropriate GENOME\_INFO input can be either a character vector or a data frame:

\*character vector\*

options:

- (1) KEGG taxonomy identifier (T0 number; e.g. "T00001")  
AND/OR KEGG organism identifier (3 or 4 letter code; e.g. "eco").
- (2) Organism's scientific name (e.g. "Escherichia coli"; case-insensitive).

Multiple strains might be matched in the KEGG organisms database and all will be processed. Strain information or full organism name can be added to reduce the search results; use the 'Definition' entry in KEGG `\url{http://www.genome.jp/kegg/catalog/org_list.html}`.

NOTE: do NOT combine NAMES with IDENTIFIERS  
 WARNING issued when there is no matching identifier in the KEGG genome/organism databases.

\*data frame\*  
 > column I - genome name/identifier (pointed to by `\code{GENOME_ID_COL}`).  
 > column II - concatenated string of either EC or K numbers using `\code{splitBy}` as delimiter (pointed to by `\code{GENES_COL}`).

KO\_in\_DEF\_EC - If the genes given are EC numbers, should K numbers present in the KEGG module definition be assumed to be present or not? Default (FALSE).

The `use_set` of argument allows users with KEGG FTP access to provide the updated data from the KEGG databases in the form of reference tables. These reference tables can be generated with the `parseKEGG` family of functions and need to have a specific format (see function descriptions for details on format).

## Value

Returns a list containing the following objects:

\$MATRIX - matrix of the datasets (rows) and the modules searched (columns) containing the fraction completeness.

\$QUERIES - data frame listing the SEARCH\_TERMS and ARGUMENTS used.

\$METADATA - data frame containing the metadata from the modules analysed (if `META_OUT == TRUE`).

Columns:

(1) MODULE_ID	(4) CLASS_I	(7) DEFINITION
(2) MODULE_NAME	(5) CLASS_II	(8) OPTIONAL
(3) NAME_SHORT	(6) CLASS_III	

An OPTIONAL entry of 'NA' indicates that there are no optional K numbers in that module.

\$ADD\_INFO - data frame containing additional information of the analysis (if `ADD_OUT == TRUE`).

Columns:

(1) GENOME_ID	(3) NAME_SHORT	(5) nBLOCKS	(7) OPTIONAL_PRESENT
(2) MODULE_ID	(4) FRACTION	(6) COVERAGE	

where 'COVERAGE' refers to the genes provided that are involved in the given module and genome.

See `misc_geneVector_module` for additional details on the output.

## See Also

`misc_geneVector_module`, `analysis_genomes_module_output`, `plot_heatmap`,  
`data_example_multi_EC_KOs`



## Examples

```
## USE T numbers
T_NUMEBERS <- paste("T0000",1:5,sep="")
OUT <- query_genomes_to_modules(T_NUMEBERS,MODULE_ID = paste("M0000",1:5,sep=""),
                                META_OUT = T, ADD_OUT = T)

## USE SPECIES NAMES
names <- c("escherichia coli","heliobacter")
OUT <- query_genomes_to_modules(names,MODULE_ID = paste("M0000",1:5,sep=""),
                                META_OUT = T, ADD_OUT = T)

## USE USER-SPECIFIED GENE SETS
data(data_example_multi_EC_KOs) # load example data set
names(data_example_multi_EC_KOs)
# "ID"      "ORG_ID"    "ORGANISM" "KOs"      "ECs"
OUT <- query_genomes_to_modules(data_example_multi_EC_KOs,GENOME_ID_COL = "ID",
                                GENES_COL = "KOs",MODULE_ID = paste("M0000",1:5,sep=""),
                                META_OUT = T,ADD_OUT = T)

# Using EC numbers (less accurate)
OUT <- query_genomes_to_modules(data_example_multi_EC_KOs,GENOME_ID_COL = "ID",
                                GENES_COL = "ECs",MODULE_ID = paste("M0000",1:5,sep=""),
                                META_OUT = T,ADD_OUT = T)
```

---

query\_missingGenes\_from\_module

*Identify missing genes from a KEGG module given a set of genes.*

---

## Description

Identify missing genes from a KEGG module given a set of genes or a genome ID to obtain a complete module.

## Usage

```
query_missingGenes_from_module(GENOME, MODULE_ID, PRINT_TO_SCREEN = TRUE,
                                use_genome_reference_table = NULL, use_module_reference_table = NULL, ...)
```

## Arguments

GENOME	- character vector containing a single genome identifier or set of genes or enzymes that define a [meta]genome. See Details.
MODULE_ID	- KEGG module ID to be analysed (e.g. "M00001").
PRINT_TO_SCREEN	- logical. Should a print-friendly result be displayed? Default(TRUE).
use_genome_reference_table	- optional. Provide a data frame with updated KEGG genome database. Default (NULL; inbuilt data used). See Details.
use_module_reference_table	- optional. Provide a data frame with updated KEGG module database OR with custom-made modules. Default (NULL; inbuilt data used). See Details.
...	- further arguments (currently unsupported)

## Details

GENOME can be a genome identifier (T0 number or a 3 or 4 letter code; e.g. "T00001" or "eco", respectively) OR a character vector containing a set of genes (i.e. either EC or K numbers; e.g. "1.1.1.1" or "K00001", respectively). Examples of the latter can be found in the following objects: `data_example_KOnumbers_vector` or `data_example_ECnumbers_vector`.

Organism name NOT supported. Use the KEGG database website ([http://www.genome.jp/kegg/catalog/org\\_list.html](http://www.genome.jp/kegg/catalog/org_list.html)) to determine the genome identifier.

Note that the pipe ('|') in the DEFINITION indicates an OR operation. This means that there are several possible genes that carry out the same reaction or function and only one is required.

The `use_` set of arguments allow users with KEGG FTP access to provide the updated data from the KEGG databases in the form of reference tables AND/OR for advanced users to provide custom-made modules (see below). These reference tables can be generated with the `parseKEGG` family of functions and need to have a specific format (see function descriptions for details on format).

The module definition (contained in `module_reference_table`) describes the relationship between genes and modules and is used to identify the modules in which gene is involved. The user can provide custom-made module definitions that use the logical expression format (however, the table format must be conserved!).

## Value

Data frame containing the following columns:

<code>BLOCK_DEF</code>	- the KEGG module DEFINITION of each block, with missing genes flagged by '*';
<code>PRESENT</code>	- binary indicator of the automatic evaluation;
<code>MISSING_GENES</code>	- list of missing genes.

## Examples

```
# Load data
data("data_example_KOnumbers_vector")
OUT <- query_missingGenes_from_module(data_example_KOnumbers_vector, "M00001")
```

---

```
query_modules_to_genomes
```

*Find a KEGG genome that has a given KEGG module complete*

---

## Description

Find a KEGG genome that has a given KEGG module complete (to a threshold). The module 'completeness' is based on the fraction of complete module blocks (given the KEGG module logical definition). Thus, a threshold of 0.5 would mean that the function would return all genomes that contain at least half of the blocks of the given module. See `parseKEGG_module` for further details on the KEGG module database.

## Usage

```
query_modules_to_genomes(MODULE_ID, threshold = 1,
  use_matrix_dataframe = NULL, use_module_reference_table = NULL, ...)
```

## Arguments

- `MODULE_ID` - KEGG module identifier (M number, e.g. "M00001").
- `threshold` - optional. Completeness fraction desired (greater or equal). Default (1). Used as `fraction >= threshold`.
- `use_matrix_dataframe` - optional. Provide module completeness fraction matrix or data frame of all KEGG genome entries with updated data. Default (NULL; inbuilt data used). See Details.
- `use_module_reference_table` - optional. Provide a data frame with updated KEGG module database OR with custom-made modules. Default (NULL; inbuilt data used). See Details.
- ... - further arguments (currently unsupported)

## Format

`use_matrix_dataframe` rows - genome identifiers, columns - module IDs.

## Details

The `use_` set of argument allows users with KEGG FTP access to provide the updated data from the KEGG databases in the form of reference tables AND/OR for advanced users to provide custom-made modules (see below). These reference tables can be generated with the `parseKEGG` family of functions and need to have a specific format (see function descriptions for details on format).

To generate `use_matrix_dataframe`, store the `$MATRIX` output of `query_genomes_to_modules` when providing all genomes using either the module default (i.e. all modules) or specifying a subset of modules AND/OR with custom-made module definitions.

## Value

If a single `MODULE_ID` is provided, a vector which contains the mfc with the KEGG genome identifiers (T number) as names is returned.

If multiple `MODULE_IDs` are provided, a matrix containing the mfc with the KEGG genome identifiers (T number) as row names and the module IDs as column names is returned.

## See Also

`parseKEGG_module`, `query_genomes_to_modules`

## Examples

```
genomes <- query_modules_to_genomes("M00001")

genomes <- query_modules_to_genomes(MODULE_ID = c("M00001", "M00002"), threshold = 0.9)
```

# Index

## \*Topic **datasets**

- data\_example\_ECnumbers\_vector, 6
- data\_example\_genomeIDs, 7
- data\_example\_KOnumbers\_vector, 7
- data\_example\_moduleIDs, 8
- data\_example\_multi\_EC\_KOs, 9
- data\_example\_sunburst, 10
- data\_example\_sunburst\_fill\_by, 10
- data\_module\_shortcode\_mapping, 11
- analysis\_genomes\_module\_output, 2, 35, 36, 44
- analysis\_genomes\_module\_output(), 34, 36
- analysis\_pca\_mean\_distance\_calculation, 4, 5
- analysis\_pca\_mean\_distance\_grouping, 3, 5, 35, 36
- analysis\_pca\_mean\_distance\_grouping(), 3, 34, 36
- data\_example\_ECnumbers\_vector, 6, 8, 46
- data\_example\_genomeIDs, 7
- data\_example\_KOnumbers\_vector, 6, 7, 46
- data\_example\_moduleIDs, 8
- data\_example\_multi\_EC\_KOs, 9, 44
- data\_example\_sunburst, 10
- data\_example\_sunburst\_fill\_by, 10
- data\_module\_shortcode\_mapping, 11
- enzyme\_reference\_table, 41
- <http://www.genome.jp/kegg/module.html>, 29
- ko\_reference\_table, 41
- misc\_axisRound, 12
- misc\_check\_duplicate\_names, 12
- misc\_create\_labels, 13
- misc\_evaluate\_block, 13
- misc\_geneVector\_module, 14, 43, 44
- misc\_module\_definition\_block\_EC, 16, 30
- misc\_module\_definition\_check, 16, 18, 30
- misc\_module\_definition\_optional, 17, 30
- misc\_module\_subgroup\_indexing, 18
- parseKEGG\_compound, 18, 21, 22
- parseKEGG\_enzyme, 19, 21, 22
- parseKEGG\_execute\_all, 21, 22
- parseKEGG\_file, 19–21, 22, 24, 26, 30, 32
- parseKEGG\_file.list, 21, 23, 27, 28
- parseKEGG\_genome, 21, 22, 23, 30, 31, 41
- parseKEGG\_ko, 21, 25
- parseKEGG\_ko\_enzyme, 21, 23, 26
- parseKEGG\_ko\_reaction, 21, 23, 27
- parseKEGG\_module, 11, 13, 15–18, 21, 22, 28, 42, 46, 47
- parseKEGG\_process\_KEGG\_taxonomy, 30
- parseKEGG\_reaction, 21, 22, 31
- plot\_heatmap, 3, 7, 8, 33, 44
- plot\_scatter, 34
- plot\_scatter\_byFactors, 3, 35
- plot\_sunburst, 10, 11, 37
- plot\_variance\_boxplot, 39
- query\_genes\_to\_genomes, 40
- query\_genes\_to\_modules, 41
- query\_genomes\_to\_modules, 3, 9, 29, 42, 47
- query\_missingGenes\_from\_module, 45
- query\_modules\_to\_genomes, 46

## Appendix F

# Bioinformatics analysis using MetQy

## F.1 Methods

### F.1.1 Hack to obtain the `genome_reference_table` object

The following code demonstrates how the `genome_reference_table` object used in Code 5.1 could be retrieved using MetQy functions without the need to have FTP access of KEGG.

Note that the `genomes` object is the same as that obtained in Code 5.1.

**Code F.1** Retrieving genome information contained in `genome_reference_table` object using MetQy functions

```
1 # Construct the KEGG genome IDs
2 > genomes <- paste("T", sprintf("%05d", 1:99999), sep="")
3 > head(genomes)
4 [1] "T00001" "T00002" "T00003" "T00004" "T00005" "T00006"
5 > tail(genomes)
6 [1] "T99994" "T99995" "T99996" "T99997" "T99998" "T99999"
7
8 # Use the query_genomes_to_modules function to get the genomic information.
9 # NOTE – one MODULE_ID has been specified to speed the computation as the default
   is to match all modules.
10 > output <- query_genomes_to_modules(genomes, MODULE_ID = "M00001")
11 # NOTE – this function evaluates whether the genome IDs are valid. It will only
   return information for those that are and a warning message will be returned
   listing the invalid genome IDs provided.
12
13 > genome_reference_table <- output$GENOME_INFO_DATA
14 > names(genome_reference_table)
15 [1] "ID" "ORG_ID" "ORGANISM" "TAXONOMY"
16
17 # Get the total number of genomes in the in—built data
18 > ncol(genome_reference_table)
19 [1] 5244
```